# INTEGRATED COMPUTER-AIDED MANUFACTURING (ICAM)
# FUNCTION MODELING MANUAL (IDEF$_0$)

## JUNE 1981

**MATERIALS LABORATORY**
**AIR FORCE WRIGHT AERONAUTICAL LABORATORIES**
**AIR FORCE SYSTEMS COMMAND**
**WRIGHT–PATTERSON AIR FORCE BASE, OHIO 45433**

This technical report has been reviewed and is approved for publication.

RICHARD R. PRESTON, Capt, USAF
Project Manager
Computer Integrated Mfg Branch

NATHAN G. TUPPER
Chief
Computer Integrated Mfg Branch

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|

| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
|---|---|---|
| UM 110231100 | | |

| 4. TITLE (and Subtitle) | 5. TYPE OF REPORT & PERIOD COVERED |
|---|---|
| INTEGRATED COMPUTER-AIDED MANUFAC- TURING (ICAM) Function Modeling Manual (IDEF$_0$) | Users Manual |
| | 6. PERFORMING ORG. REPORT NUMBER |

| 7. AUTHOR(s) | 8. CONTRACT OR GRANT NUMBER(s) |
|---|---|
| | F33615-78-C-5158 |

| 9. PERFORMING ORGANIZATION NAME AND ADDRESS | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
|---|---|
| SofTech, Inc. 460 Totten Pond Road Waltham, MA 02154 | 78011F |

| 11. CONTROLLING OFFICE NAME AND ADDRESS | 12. REPORT DATE |
|---|---|
| Materials Laboratory AF Wright Aeronautical Laboratories Wright-Patterson AFB, OH 45433 | June 1981 |
| | 13. NUMBER OF PAGES |
| | 141 |

| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | 15. SECURITY CLASS. (of this report) |
|---|---|
| | UNCLASSIFIED |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Manufacturing Architecture
Model of Manufacturing
Integrated Computer-Aided Manufacturing
Subsystem Integration

ICAM
IDEF
AUTOIDEF$_0$
MFG0
DESIGN0

MFG1
SMC2
MCMM0/MFG0
SMC0/MFG0

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This manual is the instruction manual for teaching function modeling methodology for describing manufacturing functions.

DD $_{1\ JAN\ 73}^{FORM}$ 1473    EDITION OF 1 NOV 65 IS OBSOLETE

# PREFACE

IDEF$_0$ is based upon SofTech's Structured Analysis and Design Technique (SADT). SADT concepts stem from a long history of problem-solving theoretical efforts by D. T. Ross, as influenced by the "Human-directed Activity Cell Model" approach of Dr. S. Hori, then of IITRI. The SADT method was first formalized in a joint course-preparation program partially funded by ITT. Later a subset of this material was revised and published as IDEF$_0$ under the ICAM Program.

Principal contributors to the development of IDEF$_0$ are:

| | | |
|---|---|---|
| D. T. Ross | MIT and SofTech, Inc. | Developer of original concepts. |
| S. Hori | IITRI | Developer of Cell Modeling Concepts |
| C. G. Feldmann | MIT and SofTech, Inc. | Primary author of first SADT Reference Manual and manager of first IDEF$_0$ applications projects. |
| D. E. Thornhill | SofTech, Inc. | Primary author of first formal SADT course material and application to government and commercial projects. |
| R. R. Bravoco | SofTech, Inc. | Application of SADT and IDEF to Manufacturing |
| M. F. Connor and D. A. Marca | SofTech, Inc. | Developer of revised commercial SADT materials and applications to commercial projects. |
| Judy Kornfeld | SofTech, Inc. | Developed criteria for judging quality of IDEF$_0$ models. |

# TABLE OF CONTENTS

SECTION 1

## INTRODUCTION

### 1.1 Purpose

The U. S. Air Force Program for Integrated Computer Aided Manufacturing (ICAM) is directed toward increasing manufacturing productivity through the systematic application of computer technology. The ICAM Program approach is to develop structured methods for applying computer technology to manufacturing and to use those methods to better understand how best to improve manufacturing productivity.

The ICAM Program identified a need to better communicate and analyze manufacturing for the people involved in improving productivity. To satisfy that need. the ICAM Program developed the IDEF (ICAM Definition) method to address particular characteristics of manufacturing. IDEF is comprised of three modeling methodologies which graphically characterize manufacturing:

$IDEF_0$ is used to produce a _function_ model which is a structured representation of the functions of a manufacturing system or environment, and of the information and objects which interrelate those functions.

$IDEF_1$ is used to produce an _information_ model which represents the structure of information needed to support the functions of a manufacturing system or environment.

$IDEF_2$ is used to produce a _dynamics_ model which represents the time varying behavior of functions, information and resources of a manufacturing system of environment.

Each of the three models individually or any group of models can form an "ARCHITECTURE" when the environment of system being modeled is comprised of component systems, organizations and/or technologies which must work together to accomplish the overall objective (production) of the manufacturing environment or system. The significance of the models being referred to as "ARCHITECTURES" is that they are blueprints or frameworks which define graphically the fundamental relationships - the

functional interfaces, identification of common, shared and discrete information, and dynamic interaction of resources. It is important to recognize that the IDEF models become Architectures when used to better understand, communicate and analyze not only the subject environment or system (manufacturing) but how its constituent components (system, organizations and technologies) fit together.

IDEF is a method, Architecture is a means and improving manufacturing productivity is the end which the ICAM Program is pursuing within the U. S. aerospace industry.

The following material is a discussion of the fundamental concepts, techniques and procedures regarding the use of $IDEF_0$ to produce a function model.

SECTION 2

IDEF$_0$ CONCEPTS

2.1    Background

The desire of the U. S. Air Force to reduce costs and lead times by assisting the Aerospace Industry in its modernization efforts has been evidenced in the many "Tech Mod" (Technology Modernization) programs now underway.  A similar goal, but using an industry-wide target rather than individual companies, was established under the ICAM Program.  In ICAM, the goal was to develop "generic subsystems" which could be used by a large number of companies to provide a significant upgrade to the industry as a whole.  These "subsystems" provide support for common manufacturing functions such as management of information, shop floor scheduling, and materials handling.

This ambitious goal needed a common "baseline" communication vehicle around which to plan, develop, and implement the subsystems in the individual Aerospace Companies.  This baseline was called the "Architecture of Manufacturing", since it was to provide an industry-wide "architecture" showing how industry works today and around which generic subsystems could be planned, developed, and implemented.

To develop the Architecture of Manufacturing, a "language" was needed in which to express and document current Aerospace Industry operations.  At the outset of ICAM, the Air Force issued a Request for Proposal to build the architecture.  A "cell modeling technique" was specified as the expressive language (where a "cell" was defined as a manufacturing cell, or operational unit).  To be successful, the language had to satisfy the following criteria:

- Since the architecture is to depict manufacturing, it must be able to express manufacturing operations in a natural and straightforward way.

- Since the subject is so vast and complex, it must be concise and provide a straightforward means of locating details of interest easily and quickly.

- Since it must be used by a wide audience, it must be able to communicate to a wide variety of Aerospace Industry manufacturing personnel as well as to Air Force ICAM Program Office Personnel.

- Since it must serve as a baseline for generic subsystem planning, development, and implementation, it must permit sufficient rigor and precision to insure orderly and correct results.

- Since the baseline must be developed through the cooperative effort of a large segment of the Aerospace Industry, it must include a methodology (rules and procedures) for its use that permit many diverse groups to develop architecture pieces and that permit wide-spread review, critique, and approval.

- Since the baseline must represent the entire Aerospace Industry rather than any one company or industry segment, the method must include a means of separating "organization" from "function"; that is, a common agreement cannot be achieved unless the individual company organizational differences are separated out and only the common functional thread is captured.

The cell-modeling technique selected by the Air Force was the SADT (Structured Analysis and Design Technique) originally developed in the early 1970's. The major subset of this technique used by the ICAM Program Office was later re-named "IDEF$_0$".

## 2.2    IDEF$_0$ Concepts

The IDEF$_0$ methodology has basic concepts which address each of the needs listed above. These basic IDEF$_0$ concepts are:

1.    Cell Modeling Graphic Representation. The "box and arrow" graphics of an IDEF$_0$ diagram show the manufacturing operation as the box, and the interfaces to/from the operation as the arrows entering/leaving the box. In order to be able to express real-life manufacturing operations, boxes operate simultaneouly with other boxes, with the interface arrows providing "constraints" as to when and how operations are triggered and controlled.

2.     <u>Conciseness</u>. The documentation of a manufacturing architecture must be concise to permit encompassing the subject matter. The linear, verbose characteristics of ordinary language text is clearly insufficient. The two-dimensional form provided by a blueprint-like language has the desired conciseness without losing the ability to express relationships such as interfaces, feedback, and error paths.

3.     <u>Communication</u>. There are several $IDEF_0$ concepts which are designed to enhance communications:

- Diagrams based upon very simple box and arrow graphics.

- English textual labels to describe box and arrow meaning, as well as glossary and text to define precise meaning of diagram elements.

- Gradual exposition of detail, featuring a hierarchy with major functions at the top and successive levels of sub-functions revealing well-bounded detail breakout.

- A "node chart" provides a quick index for locating details within the hierarchic structure of diagrams.

- Limitation of detail on each successive diagram to not more than six sub-functions for ease in reader comprehension.

4.     <u>Rigor and Precision</u>. The rules of $IDEF_0$ require sufficient rigor and precision to satisfy ICAM architecture needs without overly constraining the analyst. $IDEF_0$ rules include:

- Detail exposition control at each level (3-6 box rule).

- Bounded Context (no omissions or additional out-of-scope detail).

- Diagram Interface Connectivity (Node Numbers, Box Numbers, C-Numbers, and Detail Reference Expression (DRE).

- Data Structure Connectivity (ICOM codes and use of parentheses).

- Uniqueness of labels and titles (no multiple names).

- Syntax Rules for Graphics (boxes and arrows).

- Data Arrow Branch Constraint (Labels for constraining data flow on branches).

- Input vs. Control Separation (Rule for determining role of data).

- Data Arrow Label Requirements (minimum labeling rules).

- Minimum Control of Function (all functions require at least one control).

- Purpose and Viewpoint (all models have a purpose and viewpoint statement).

5. <u>Methodology</u>. Step-by-step procedures are provided for modeling, review, and integration tasks. Formal courses for transferring the methodology are available for training Aerospace Industry personnel in these procedures.

6. <u>Organization vs. Function</u>. The separation of organization from function is included in the purpose of the model, and carried out by the selection of functions and interface names during model development. This concept is taught in the $IDEF_0$ course, and continual review during model development ensures that organizational viewpoints are avoided.

## 2.3 Discussion of Individual $IDEF_0$ Concepts

In the remaining sub-sections descriptions of some of the basic concepts are elaborated to clarify them and show their utility in ICAM.

### 2.3.1 Cell Modeling Graphics

The $IDEF_0$ methodology may be used to model a wide variety of "systems", where a system may include any combination of hardware, software, and people. For new systems $IDEF_0$ may be used first to specify the requirements and functions and then to design an implementation that meets the requirements and performs the functions. For existing systems, $IDEF_0$ can be used to analyze the functions the system performs and to record the mechanisms by which these are done.

The result of applying IDEF$_0$ is a model. A <u>model</u> consists of diagrams, texts, and glossary, cross-referenced to each other. Diagrams are the major components of a model. All manufacturing functions and interfaces are represented as boxes (functions) and arrows (interfaces) on diagrams.

The position at which the arrow enters a box conveys the specific role of the interface. The manufacturing controls enter the top of the box, whereas the materials or information acted upon by the manufacturing operation enter the box from the left, resulting in the output of the operation, which leaves the right-hand side of the box. The mechanism (person or automated system) which performs the operation enters the bottom of the box (see Figure 2-1).
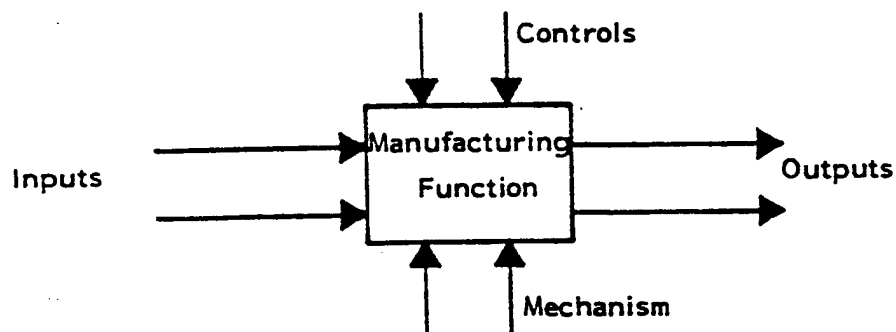


Figure 2-1. Function Box and Interface Arrows

These box and arrow meanings are used to relate several sub-functions on a diagram comprising a more general function. This diagram is a "constraint diagram" which shows the specific interfaces which constrain each sub-function, as well as the sources and targets of the interface constraints (see Figure 2-2).
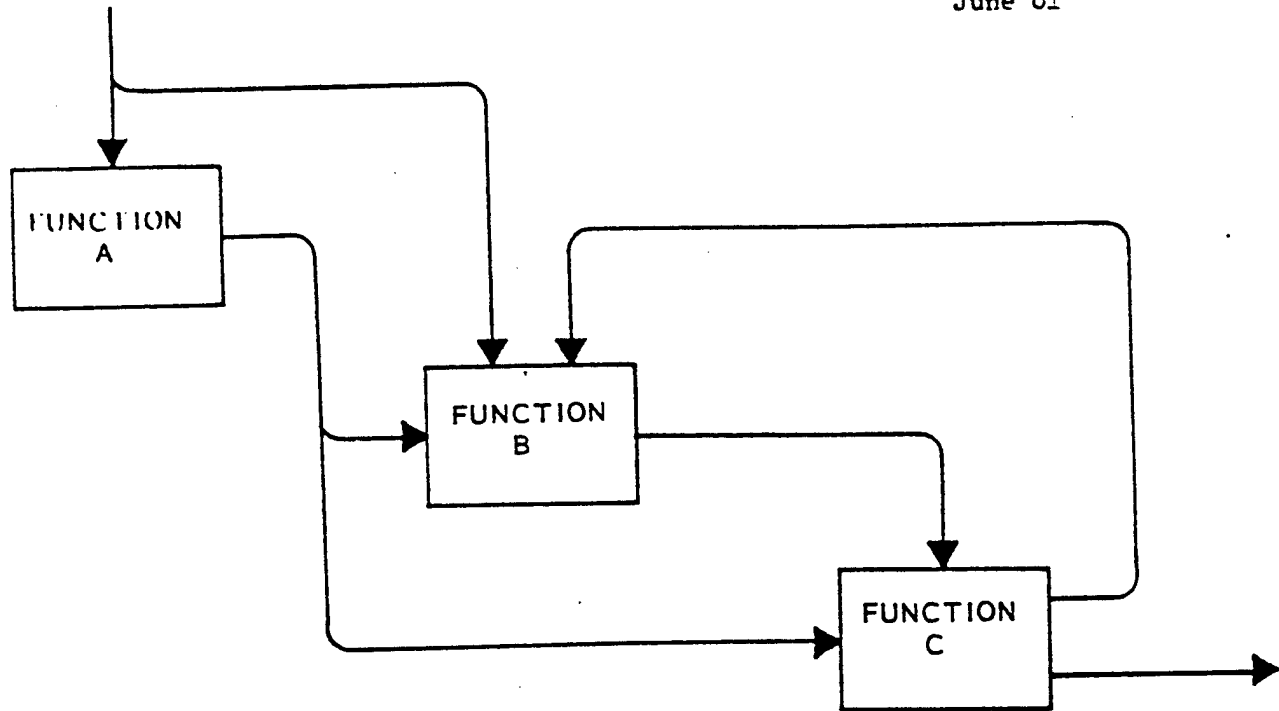
Figure 2-2. Constraint Diagrams

(Function B is constrained by one input and two controls, and
produces a single output, which constrains Function C).

Here, the term "constrains" means that a function uses the material
or information shown entering the box, and, therefore, is constrained from
operating by the interface:  the function cannot act until the contents of
the interface arrow is provided, and the way in which the function operates
depends upon the details (value, number, etc.) of the contents of the
interface arrow.

## 2.3.2  Communication by Gradual Exposition of Detail

One of the most important features of $IDEF_0$ is that it gradually
introduces greater and greater levels of detail through the diagram
structure comprising the model.  In this way, communication is enhanced
by providing the reader with a well-bounded topic with a manageable
amount of new information to learn from each diagram.

The structure of an $IDEF_0$ model is shown in Figure 2-3.  Here, a
series of four diagrams is shown with each diagram's relation to the others.

MORE GENERAL

MORE DETAILED

This diagram is
the "parent" of
this diagram.
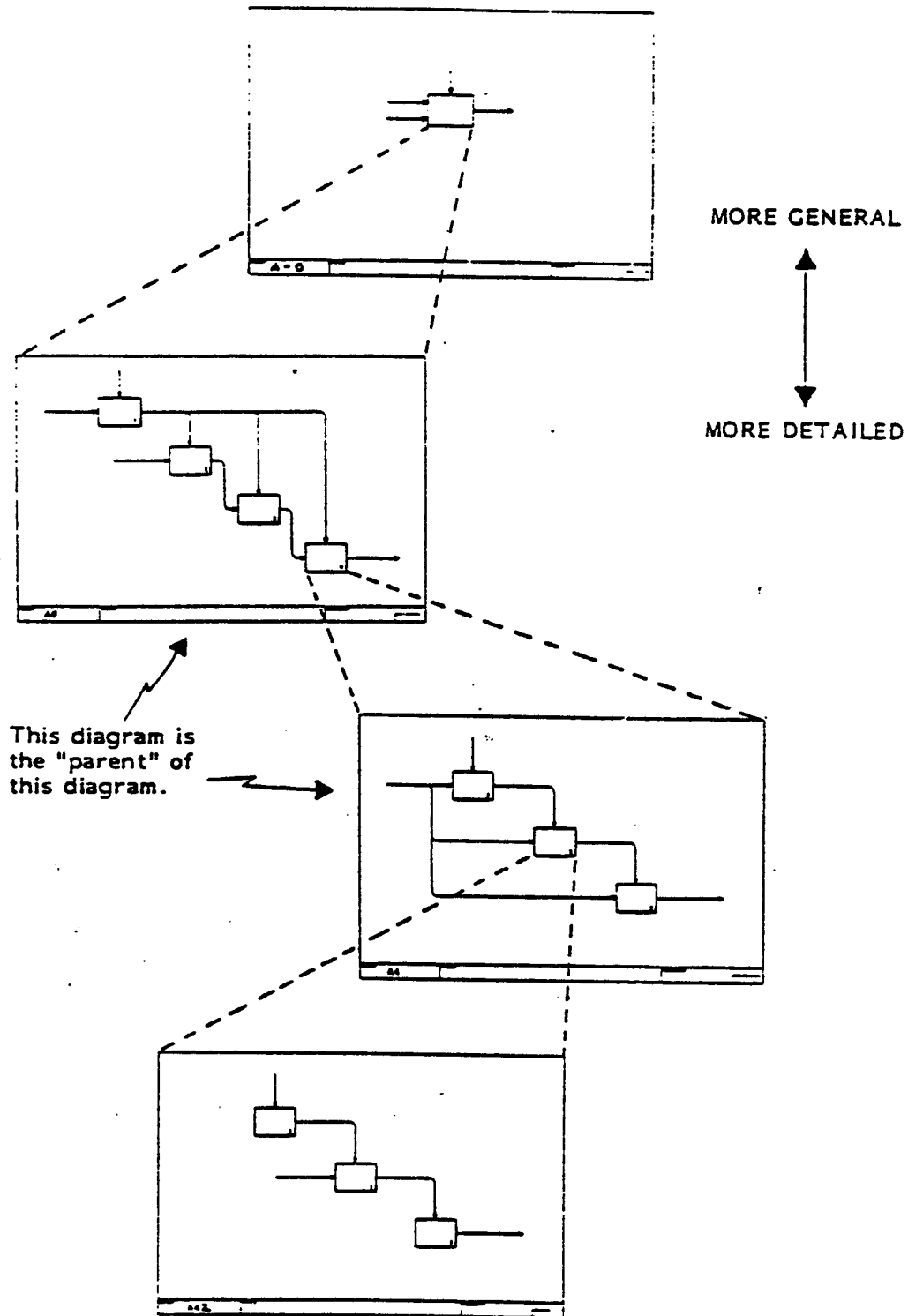
EVERY COMPONENT MAY BE DECOMPOSED IN ANOTHER DIAGRAM

EVERY DIAGRAM SHOWS THE "INSIDE" OF A BOX ON A PARENT DIAGRAM

Figure 2-3.  IDEF$_0$ Model Structure

An IDEF$_0$ model starts by representing the whole system as a simple unit -- a box with arrow interfaces to functions outside the system. Since the single box represents the system as a whole, the descriptive name written in the box is general. The same is true of the interface arrows since they also represent the complete set of external interfaces to the system as a whole.

The box that represents the system as a single module is then detailed on another diagram with boxes connected by interface arrows. These boxes represent major sub-functions (submodules) of the single parent module. This decomposition reveals a complete set of submodules, each represented as a box whose boundaries are defined by the interface arrows. Each of these submodule boxes may be similarly decomposed to expose even more detail.

IDEF$_0$ provides rules to introduce further detail during decomposition. A module is always divided into no fewer than three, but no more than six submodules. The upper limit of six forces the use of a hierarchy to describe complex subjects. The lower limit of three insures that enough detail is introduced to make the decomposition of interest.

Each diagram in a model is shown in precise relationship to other diagrams by means of interconnecting arrows. When a module is decomposed into submodules, the interface between the submodules are shown as arrows. The name of each submodule box plus its labeled interfaces define a bounded context for that submodule.

In all cases, every submodule is restricted to contain only those elements that lie within the scope of its parent module. Further, the module cannot omit any elements. Thus, as already indicated, the parent box and its interfaces provide a context. Nothing may be added or removed from this precise boundary.

## 2.3.3  Disciplined Teamwork

The IDEF$_0$ methodology includes procedures for developing and critiquing models by a large group of people, as well as integrating support subsystems into an IDEF$_0$ Architecture. Additional supporting procedures such as librarian rules and procedures, are also included in the IDEF$_0$ methodology. (It should be noted that some of these rules and procedures, such as the Kit Cycle critique procedure, are also used with other IDEF techniques.)

The creation of an IDEF$_0$ model is the most basic of these "disciplined teamwork" procedures. The creation of a model is a dynamic process which usually requires the participation of more than one person. Throughout a project, authors create initial diagrams which are distributed to project members for review and comment. The discipline requires that each person expected to make comments about a diagram shall make them in writing and submit them to the author of the diagram. This cycle continues until the diagrams, and eventually, the entire model, are officially accepted.

IDEF includes procedures for retaining written records of all decisions and alternate approaches as they unfold during the project. Copies of the diagrams created by an author are critiqued by knowledgeable commenters who document suggestions directly onto the copies. Authors respond to each comment in writing on the same copy. Suggestions are accepted or rejected in writing along with the reasoning used. As changes and corrections are made, outdated versions of diagrams are retained in the project files.

The diagrams are changed to reflect corrections and comments. More detail is added to the model by the creation of more diagrams which also are reviewed and changed. The final model represents an agreement on a representation of the system from a given viewpoint and for a given purpose. This representation can be easily read by others outside the initial project, used for presenting the system definition in short stand-up briefings or in week long walkthroughs, and for organizing new projects to work on system changes.

SECTION 3

# UNDERSTANDING IDEF$_0$ DIAGRAMS

## 3.0 Introduction

A model is a representation of a system. It may describe what a system is, what it does, or what things it works on.

Systems are composed of interfacing or interdependent parts that work together to perform a useful function. The parts may be anything, including machinery, information, objects, processes, software, or people. IDEF$_0$ can be used to describe functions performed by systems or parts of systems and the information or things through which the functions interface.

IDEF$_0$ represents a system by means of a model composed of diagrams, text, and glossary. Diagrams are composed simply of boxes and arrows. In these diagrams, the boxes represent activities and arrows represent things processed by the system.

## 3.1 IDEF$_0$ Symbols

### 3.1.1 Diagrams

A model is a series of diagrams with supportive documentation that break a complex subject into its component parts. The initial diagram is the most general or abstract description of the whole system. This diagram shows each major component as a box. The details of each major component are shown on other diagrams as boxes. These boxes can be broken down into still more diagrams, until the system is described to any desired level of detail.

Each detailed diagram is the decomposition of a box on a more general diagram. At each step, the general diagram is said to be the "parent" of the detailed diagram. A detailed diagram is best thought of as fitting "inside" a parent box. (See Figure 3-1.)

MORE GENERAL

MORE DETAILED

This diagram is
the "parent" of
this diagram.
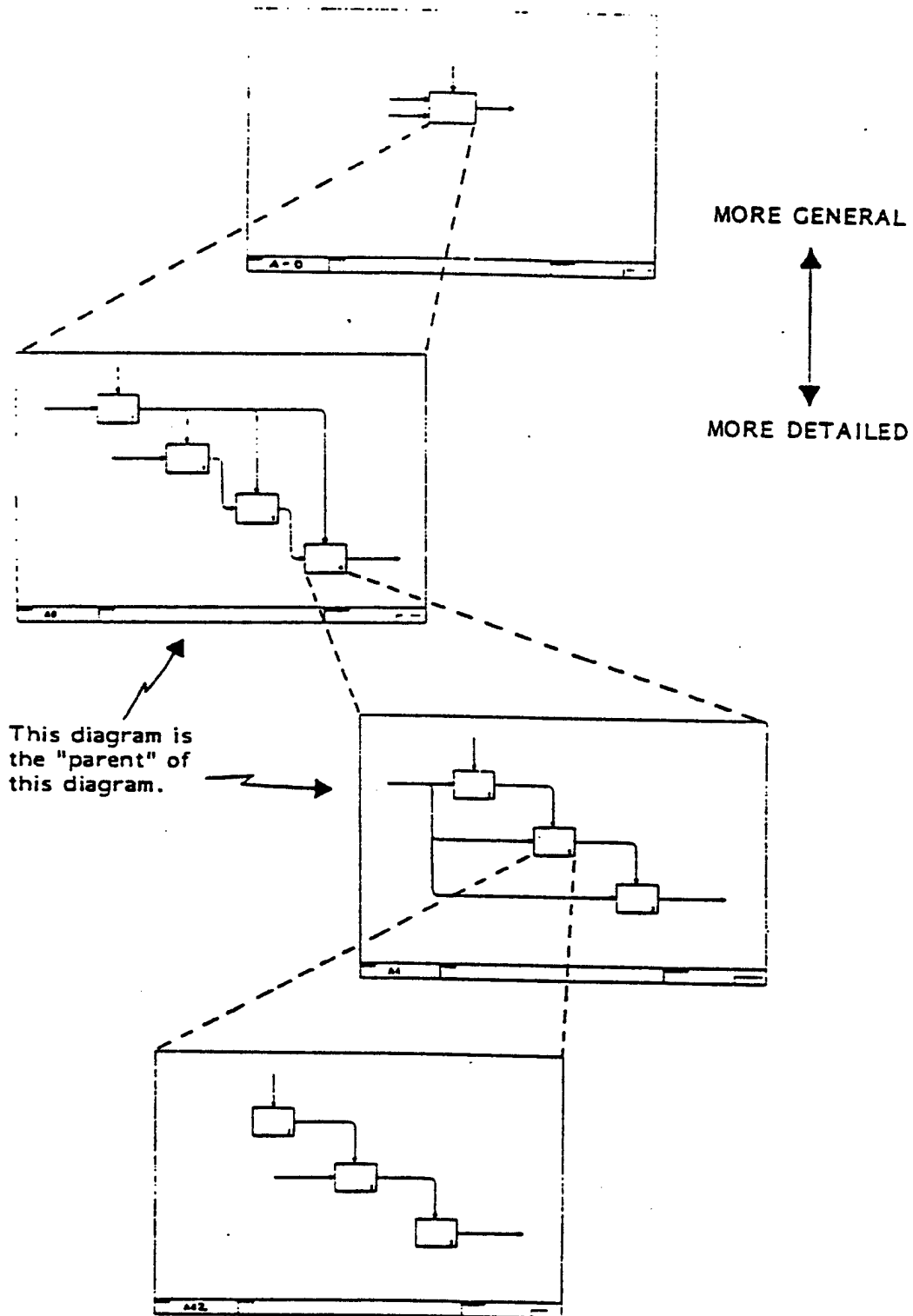
Figure 3-1.   Decomposition of Diagrams

Boxes represent system functions (activities, actions, processes or operations) and arrows represent data (either information or objects). A box on the upper diagram is detailed by the boxes and arrows of the lower diagram. Arrows entering and leaving the upper <u>box</u> are exactly those arrows entering and leaving the lower <u>diagram</u>, because both the box and the diagram represent exactly the same part of the system.

A fundamental principle of IDEF$_0$ is that a diagram cannot have fewer than three nor more than six boxes. This approach ensures uniform, systematic exposition of successive levels of detail. The upper limit of six was chosen since physchological experiments have shown that it is difficult to grasp more than 5-7 distinct concepts at one time. The lower limit of three was chosen to insure that enough detail is introduced to make the decomposition meaningful. High-level diagrams encompass a wide range of detail so that words on the boxes and arrows must be abstract and describe general concepts. Successive diagrams at lower levels gradually reveal this detail, using more specific terms, one step at a time.

### 3.1.2 <u>Boxes</u>

Boxes on a diagram represent functions. Functions show what must be accomplished without identifying any other necessary aspect such as needs or means. Functions are described by an active verb phrase written inside the box. Each box on a diagram is numbered in its lower right corner, in order from "1" to at most "6."

<u>A function is anything that can be named with an active verb phrase</u>. This includes everything from the concrete to the conceptual, such as:

| | | | |
|---|---|---|---|
| tighten | assemble | classify | adapt |
| attach | transcribe | construct | resolve |
| measure | evaluate | solve | develop |

Such functions occur over periods of time. Functions are not expressed as nouns, such as "maintenance."

### 3.1.3 Box/Arrow Relationship

The arrows that connect to a box represent objects or information needed by or produced by the function. They are each labeled with a noun phrase, written beside the arrow. "Data" may be information, objects, or anything that can be described with a noun phrase. The arrows are constraints that define the boxes, not sequences or flows of functions (Figure 3-2).

The side of the box at which an arrow enters or leaves shows the arrow's role as an input, a control or an output. Incoming arrows (left and top of box) show the data needed to perform the function. Outgoing arrows (right of box) show the data created when the function is performed. From left to right (input to output), a function transforms data. An input is converted by the function into the output. The terms input and output convey the notion that a box represents a transition from a "before" to an "after" state of affairs.
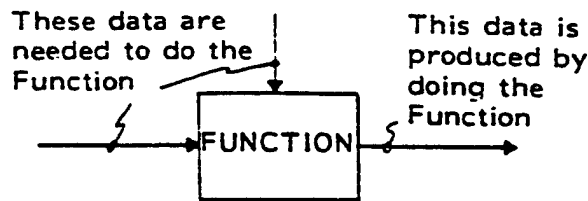


**Figure 3-2. Arrows Clarify and Bound the Meaning of Each Box**

A control describes the conditions or circumstances that govern the function. The roles of input and control are different. The distinction is important to understanding the operation of systems. The assumption is that "an arrow is a control unless it obviously serves only as input." Every function box will have at least one control arrow.

The bottom of a box is reserved to indicate a <u>mechanism</u>, which may be the person or device which carries out the function. The input and output shows WHAT is done by the function, the control shows WHY it is done, and the mechanism shows HOW it is done. (Figure 3-3).

Figure 3-3. Box/Arrow Relationship

Boxes represent collections of related functions, not just monolithic actions. A box may perform various parts of its function under different circumstances, using different combinations of its input and controls and producing different outputs. These are called the different <u>activations</u> of the box. There may be several arrows on any one side of a box which may indicate different activations.

If it is unclear whether a particular word is a noun (data) or a verb (function), an "(n)" or "(v)" may be appended to the word. For example, the word "Record" could mean a record, or the action of recording. "Record (n)" is used for the former case, and "Record (v)" is used for the latter.

The arrows on an $IDEF_0$ diagram represent data constraints. They do not represent flow or sequence. Connecting the output of one box to the input or control of another box shows a <u>constraint</u>. (Figure 3-4). The box receiving the data is constrained, since the function cannot be performed until the data is made available by the box that produces it. The arrows entering a box show all the data that is needed for the function to be performed.

Figure 3-4. Meaning of Constraint

Several functions on a single diagram could be performed simultaneously, if the needed constraints have been satisfied. Arrows connect boxes, and an output of one box may provide some or all of the data needed by one or more other boxes. (Figure 3-5).



Figure 3-5. Simultaneous Action

3-6

Neither sequence nor time is explicit in $IDEF_0$ diagrams. Feedback, iteration, continuous processes, and overlapping (in time) functions are easily shown by arrows. (Figure 3-6). The arrows connecting the left ("input") or top ("control") of a box are constraints. They represent data or objects needed to perform some part of the function. For example, a draft system specification submitted for review may be approved and thus become final, or it may be returned with comments and with a request that a new draft be submitted. The latter reactivates the design function. Both the design and the review are done with respect to the system requirements.
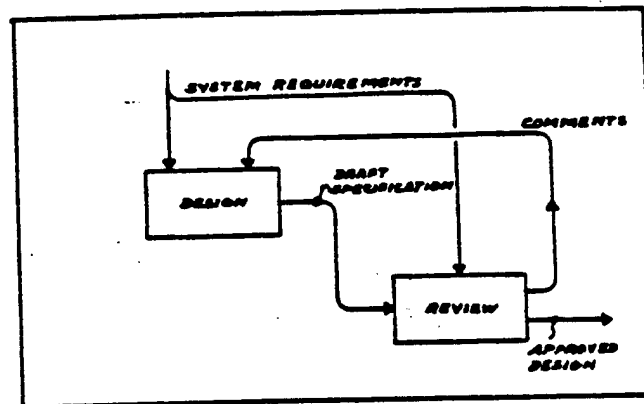


Figure 3-6. Example of Feedback

Arrows may branch (implying that the same data is needed by more than one function) or they may join (implying that the same class of data may be produced by more than one function). The branches may each represent the same thing, or different things of the same general type. Labels state what the arrows represent. The labels on branches and joins provide a detailing of the content of the more general arrow just as lower level diagrams provide detailing of boxes.

Data arrows, like function boxes, represent categories. It is useful to think of high level data arrows as "pipelines' or "conduits." High level arrows have general labels. If they branch, each branch will have a more specific label. (Figure 3-7). Arrow labels must convey the author's intent to the reader. Using fewer arrows will reduce clutter and make a diagram easier to understand but requires careful choice of meaningful words to convey the message.
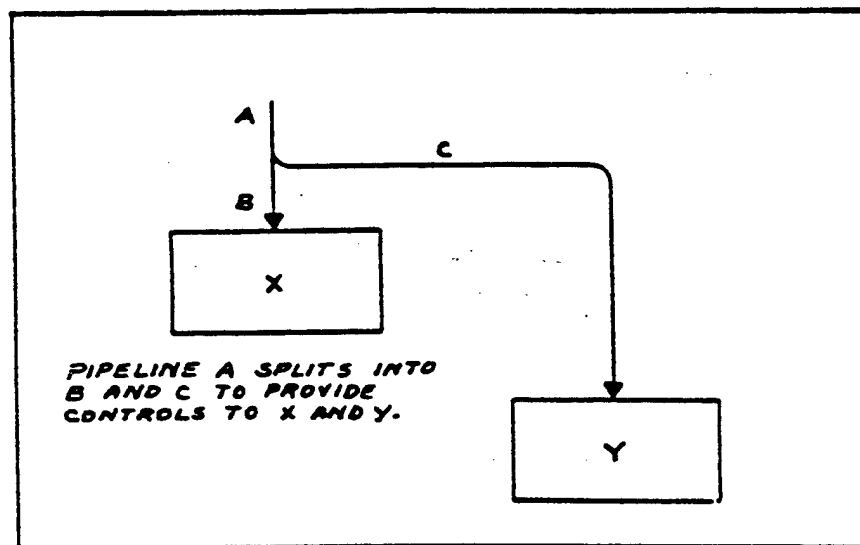
A

C

B

X

PIPELINE A SPLITS INTO
B AND C TO PROVIDE
CONTROLS TO X AND Y.

Y

Figure 3-7. Example of Arrow Branching

On any given diagram, data may be represented by an internal arrow (both ends connected to boxes shown on the diagram) or a boundary arrow (one end unconnected, implying production by or use by a function outside the scope of the diagram).

## 3.1.3.1 Arrow Connections Between Boxes

To form a complete diagram, from three to six boxes are drawn and connected by input, output, and control arrows. Any output arrow may provide some or all of the input or control (or mechanism) to any other box. All manner of arrow branches and joins are used to show box relationships. An output arrow may branch and provide data to several boxes. Arrows that are unconnected at one end represent data that is supplied or consumed outside the scope of the diagram.

This branch means that "files" (produced by box 1) are composed of "customer records" (needed by box 2) and "Price and Tax Tables" (needed by box 3).

This join means that "account entries" are created by some from DELIVER PRODUCTS (box 2) and/or some from DO BILLING (box 3).



This chain of input and output arrows means that "orders," upon delivery (box 2), are recorded as "transactions," which, when billed (box 3), are reflected on "invoices."

Figure 3-8. Arrow Connections

## 3.1.3.2 Mechanism Arrows

A box represents a function. The input data (on the left) are transformed into output data (on the right). Controls (on the top) govern the way the function is done. Mechanisms (on the bottom) indicate the means by which the function is performed. (Figure 3-9). A "mechanism" might be a person or a computer or a machine or some other device which aids in carrying out the function. The box itself, with its inputs, controls, and outputs indicates WHAT the system does. The mechanism shows HOW that function is accomplished. Diagrams drawn without mechanisms show what functions a system must perform. Mechanisms will specify how those functions are to be performed.
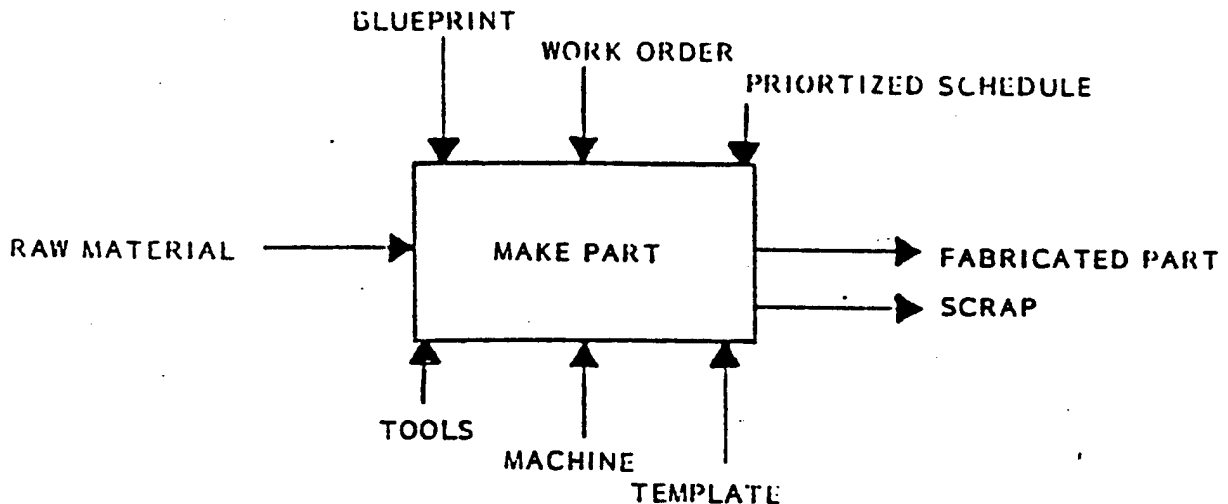
Figure 3-9. Example of Mechanism

A downward-pointing mechanism arrow (known as a "call") indicates a "system" that completely performs the function of the box. If there is a need for further detailing, it will be found in a separate model of the mechanism itself. (Figure 3-10).
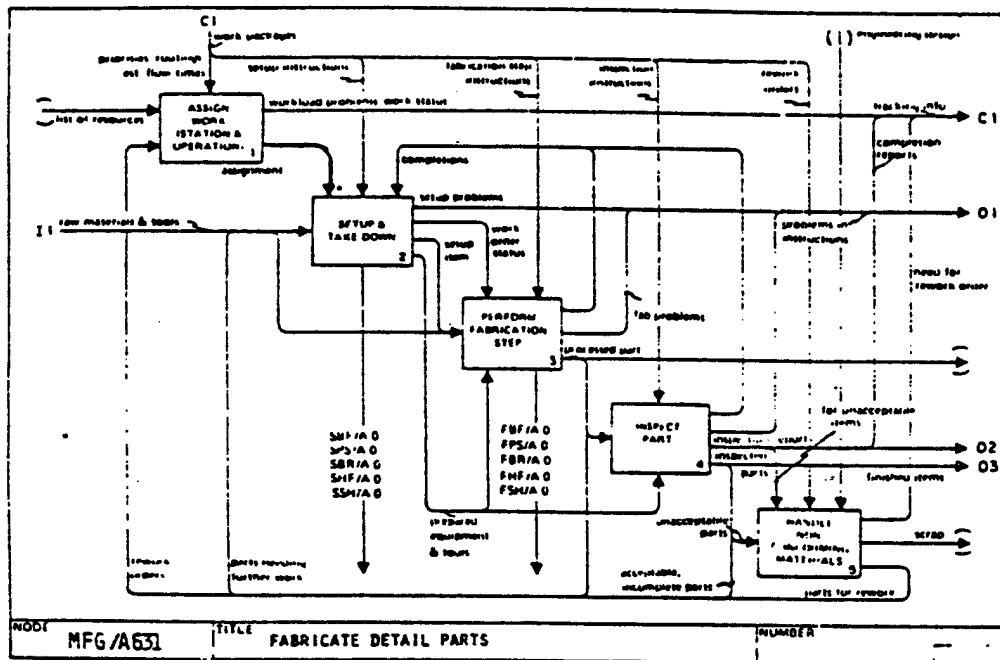
Figure 3-10.   Illustration of Mechanism Model Calls

Mechanism arrows may be the output of other boxes, if those functions create or prepare devices from their inputs and controls. (Figure 3-11).
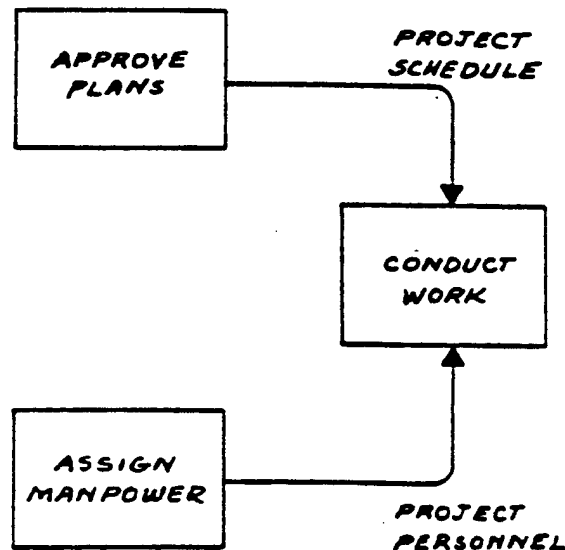


Figure 3-11.   Example of an Output that Becomes a Mechanism

### 3.1.4  An Example of an IDEF$_0$ Diagram

Figure 3-12 is a complete IDEF$_0$ diagram taken from the Function Model of "Manufacture Product (MFG0)."

The three boxes of A612 (Figure 3-13) are a decomposition of Box 2 of A61 "Control Production Orders." Based on production requirements and existing shop loads, an expected load is forecast (Box 1 of A612). Given the 1) expected load, 2) shop's capacity, and 3) specific stop orders, adjustments to the schedule are identified in Box 2. The schedule is then actually adjusted (Box 3).

Note that a single box may perform its function under different circumstances. Box 1 may occur even if there are no "revised order release dates" provided by Box 3. Or it may occur when "revised order release dates" are provided, even though "production requirements" and "shop load status" have not changed. These different occurrences are known as different activations of a box. They may be formally specified but in many cases will be naturally and intuitively understood by anyone familiar with the box and arrow notation. For example, Box 2 can occur even if "stop orders and material shortage reports" are absent. Also Box 3 does not produce "problems" with every occurrence.
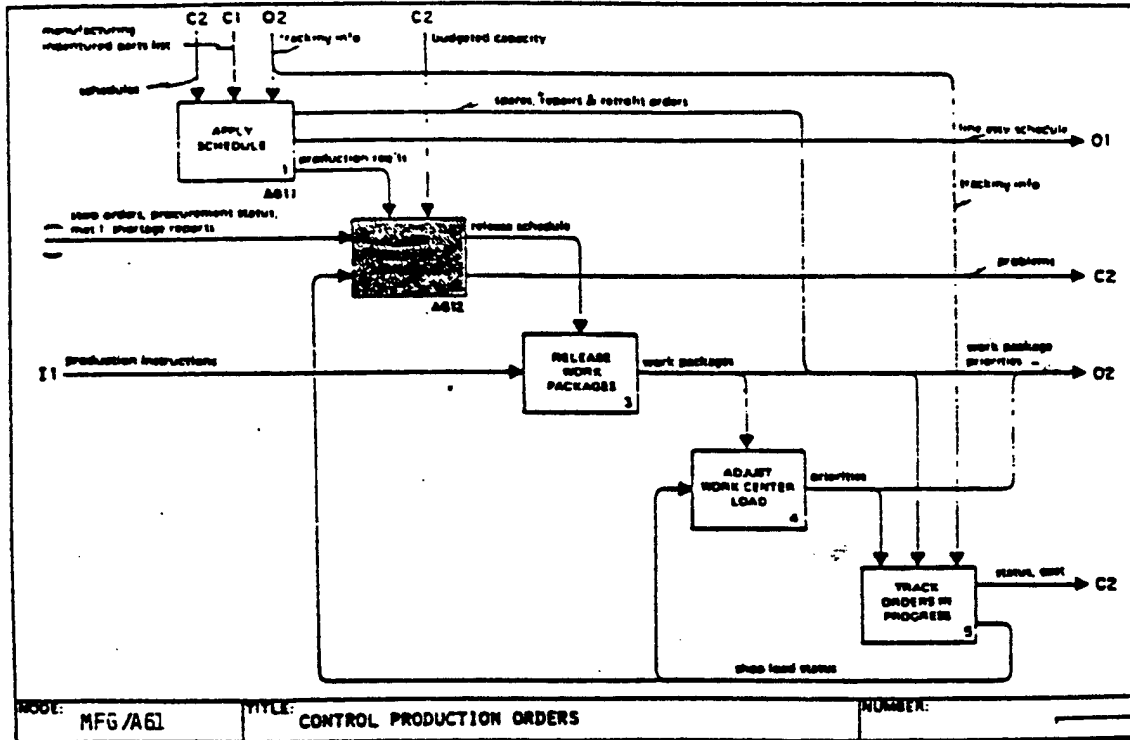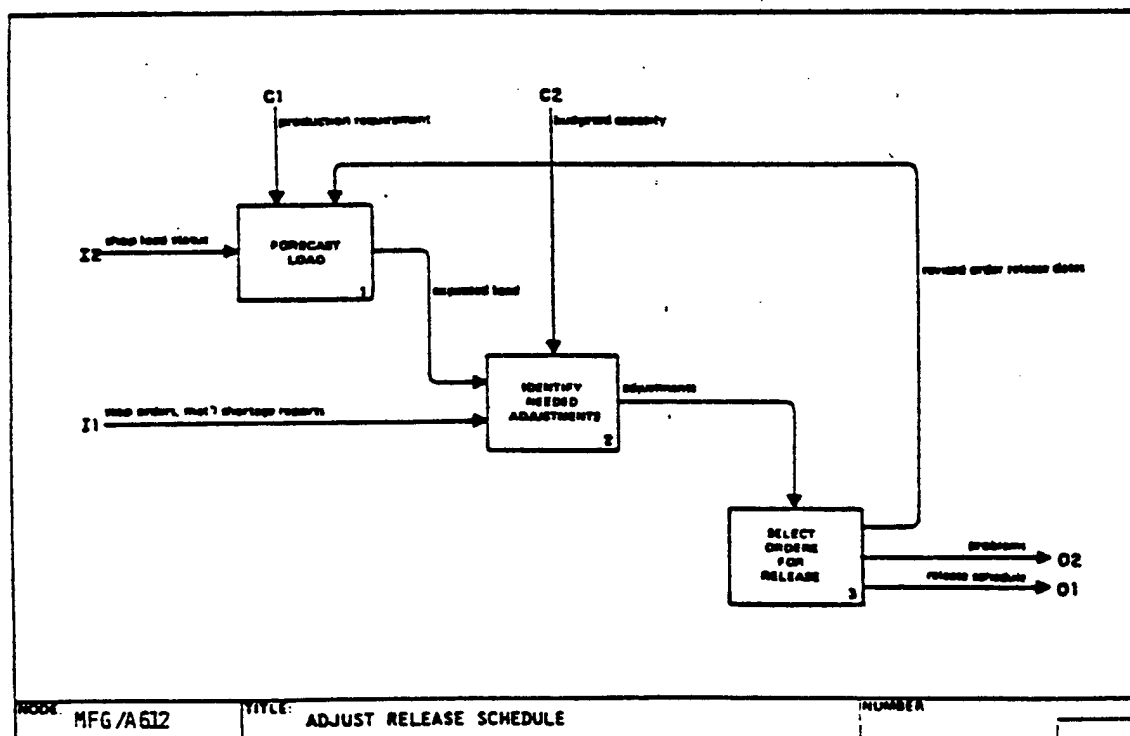
Figure 3-12.  Control Production Orders



Figure 3-13.  Decomposition of Box 2, Diagram A61

## 3.2  Additional Symbols

Further notation is needed to structure diagrams so that they form a coherent, consistent model.

To create a model composed of diagrams, FEO's, text, and glossary, we need to:

- indicate the position of each diagram in a model and the supportive documentation associated with each diagram.  This will be done with <u>reference expressions</u>;

- indicate the connection of boundary arrows to arrows of the parent diagram.  This will be done with <u>ICOM codes</u>;

- suppress unnecessary detail.  This will be done with <u>tunnelled arrows</u>.

These conventions fill out the set of symbols that enable diagrams to accurately reflect real system functions.

## 3.2.1  Reference Expressions

## 3.2.1.1  Node Numbers

As explained in Section 3.1.1, each diagram is limited to three to six boxes.  Each box on a diagram is numbered.  A box on any diagram may be further described by a "lower" diagram which may also be further detailed in more diagrams.  This forms a <u>hierarchy</u> of diagrams.

<u>Node numbers</u> are used to indicate the position of any diagram or box in the hierarchy.  For example, A21 is the diagram that details box 1 on the A2 diagram.  Similarly, A2 details box 2 on the A0 diagram which is the top-most complete diagram of a model.  This hierarchy may be shown in a chart of diagram names and node numbers called a <u>node tree</u>.  Figure 3-14 is a typical node tree.
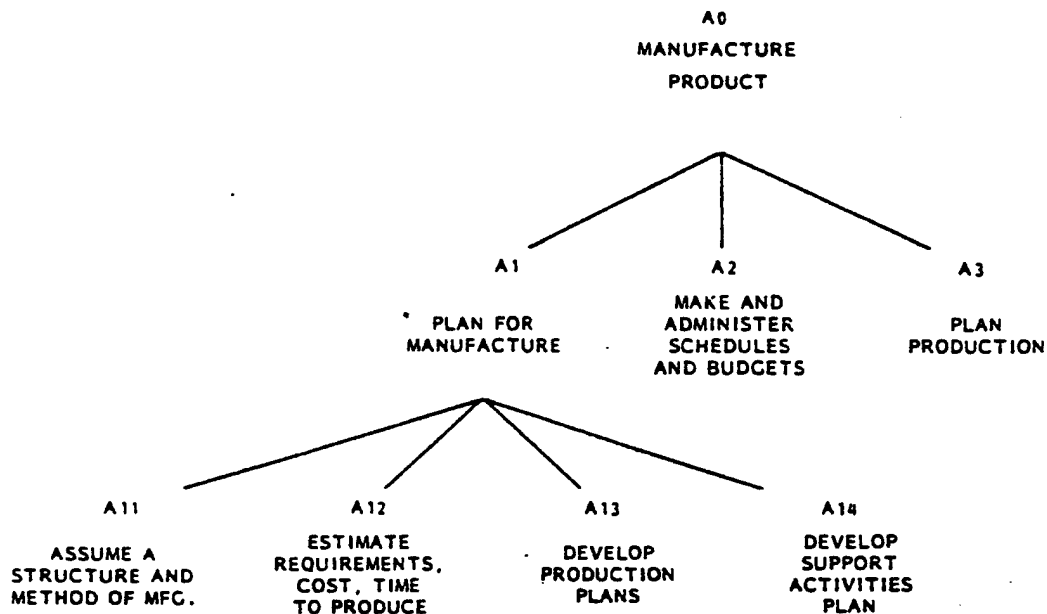
A0
MANUFACTURE
PRODUCT

A1
PLAN FOR
MANUFACTURE

A2
MAKE AND
ADMINISTER
SCHEDULES
AND BUDGETS

A3
PLAN
PRODUCTION

A11
ASSUME A
STRUCTURE AND
METHOD OF MFC.

A12
ESTIMATE
REQUIREMENTS,
COST, TIME
TO PRODUCE

A13
DEVELOP
PRODUCTION
PLANS

A14
DEVELOP
SUPPORT
ACTIVITIES
PLAN

Figure 3-14.   Diagrams form a "Hierarchy" shown by a Node Tree

All node numbers of $IDEF_0$ diagrams begin with the letter A, which identifies them as "Activity" or function diagrams.   A one-box diagram is provided as the "context" or parent of the whole model.   By convention, the diagram has the node number "A-0" (A minus zero).   If a full diagram is provided to make the context of the A-0 complete, that is numbered "A-1."   This can, if necessary, proceed upward.   Some complex subjects have actually started with "A-4," even though A-0 is still the "top" of the model.   An example of this can be found in the Function Model of "Manufacture Product" (MFG0), Volume VII of the Integrated Computer Aided Manufacturing (ICAM) Architecture Part II final Technical Report AFWAL-TR-81-4023.

Other node numbers are sometimes used. A "FEO" ("For Exposition Only") diagram is any diagram that falls outside the strict hierarchy. FEO's may contain more than six boxes, partial arrow structures, or anything needed by an author to illustrate a point. Node numbers for FEO's contain "F" (e.g., A2F). Glossary definitions support diagrams. Node numbers for a glossary contain "G" (e.g., A1G). Text numbers contain "T" and follow the node number of the diagram with which they are associated (e.g., A2T). There may be more than one FEO or glossary (e.g., A2F1, A2F2, ...), but there should be no more than one page of text associated with a given diagram. (Text, FEO's and Glossary are explained in Section 6.)

Node numbers are also used to indicate the decomposition of a box in a diagram. If a box has been decomposed, the node number of the diagram which represents the decomposition is written outside the box under the right hand corner. In Figure 3-15, the reference expression under boxes 1 and 2 indicate that they have been decomposed.

## 3.2.1.2 Model Names and Node Numbers

Each model has a name, which should be chosen for maximum clarity to distinguish it from other models. For example:

TOPIC

Diagrams in the model are referred to by adding a slash and the node number to the model name. For example:

TOPIC/A3

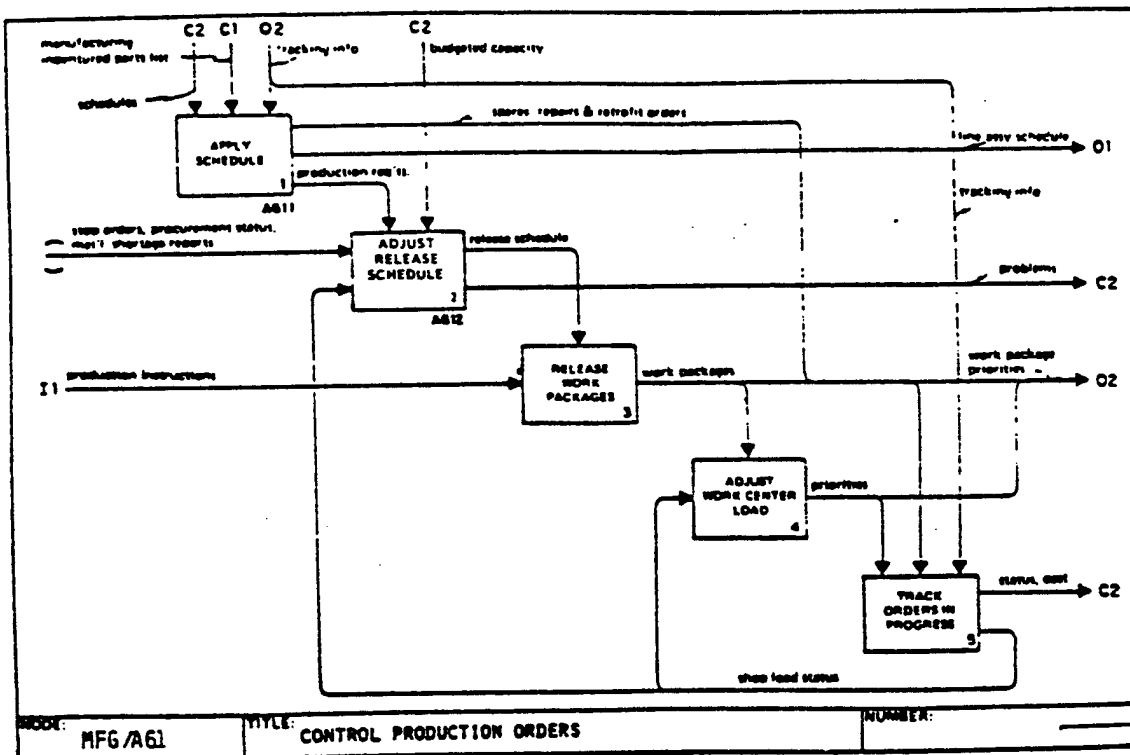It is this form which is usually written as the complete node number of each diagram of a model.

Figure 3-15. Reference Expressions

## 3.2.2 Continuing Arrows Across Diagram Boundaries

Some arrows are connected at both ends to boxes on the same diagram and other arrows have one end unconnected. (Figure 3-16). The unconnected arrows represent inputs, controls, or outputs of the parent box. The source or destination of these boundary arrows can only be found by examining the parent diagram.
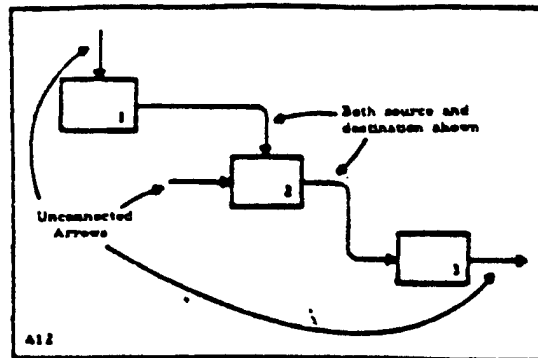
Figure 3-16.   Boundary Arrows are Derived from the Parent Diagram

The unconnected ends must match the arrows on the parent diagram.   All such boundary arrows must continue on the parent to make the model complete and consistent.
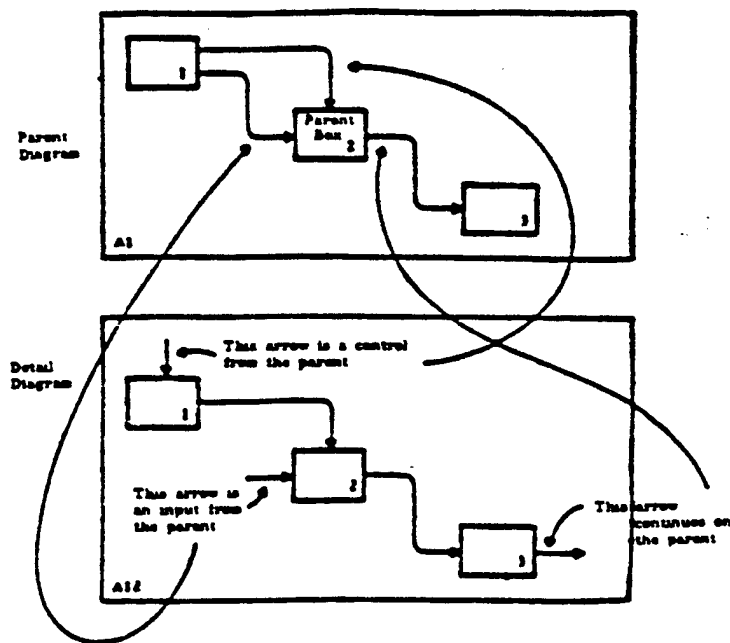


Figure 3-17.   The match must be complete and consistent.

Boundary arrows at the A-0 level are called external arrows because the A-0 diagram establishes the context of the model and all arrows related between the model and that which is external to the context or scope of the model.

### 3.2.3  Coding Boundary Arrows

A specific notation, called ICOM codes, specifies the matching connections. The letter I, C, O, or M is written near the unconnected end of each boundary arrow on the detail diagram. This identifies that the arrow is shown as an Input, Control, Output, or Mechanism on the parent box. This letter is followed by a number giving the position at which the arrow is shown entering or leaving the parent box, numbering left to right and top to bottom. For example, "C3" written on an arrow in the detail diagram indicates that this arrow corresponds to the third control arrow entering the parent box.

This coding relates each diagram to its own parent. New codes are assigned when the detail diagram becomes a parent diagram through decomposition. Using this letter-number matching scheme, an arrow shown as control or as input on a parent diagram is not limited to the same role on a detail diagram. In Figure 3-18, C2 on the parent box appears as an input to box 1 on its detail diagram. ICOM codes must be written at the unconnected ends of all boundary arrows except for the very topmost diagram in a model, and on tunneled arrows.
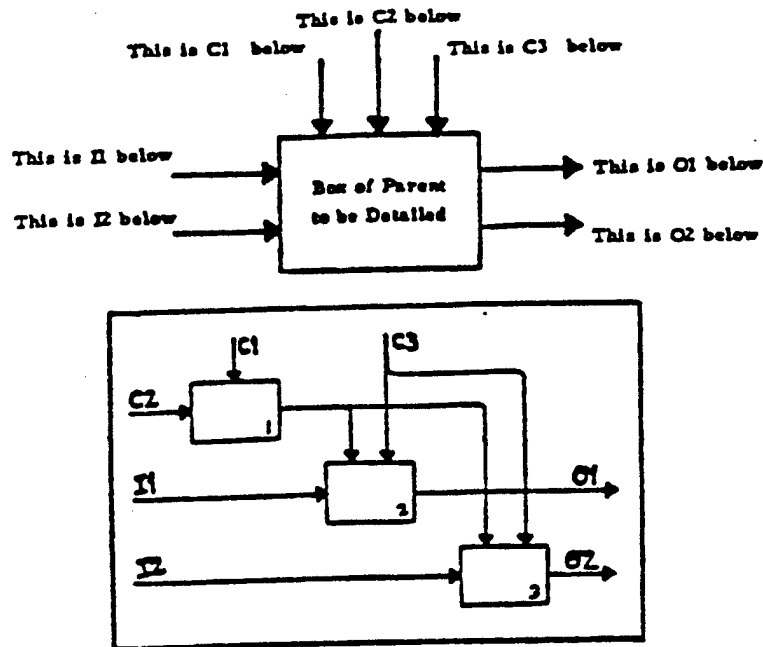
This is C2 below

This is C1 below        This is C3 below

This is I1 below          Box of Parent        This is O1 below
                         to be Detailed

This is I2 below                                This is O2 below

**Figure 3-18.   Codes are Written on the Detail Diagram**

## 3.2.4  Tunnelled Arrows

Tunnelled arrows indicate that the data conveyed by these arrows was not relevant to a particular level of detail.
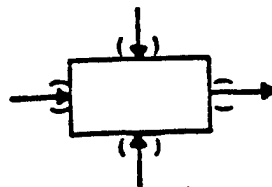
**Figure 3-19.   Tunnelled Arrows at Connected End**

Tunnelling an arrow where it connects to a box (Figure 3-19) indicates that the data conveyed is not necessary at the next level of decomposition.
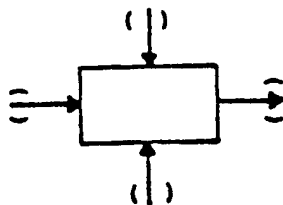
Figure 3-20.  Tunnelled Arrows at Unconnected Ends

Tunnelling an arrow at the unconnected end (Figure 3-20) indicates that the data conveyed is not relevant to or supplied by the parent diagram.

Parenthesizing the unconnected ends says, "This arrow does not appear in the parent diagram.  It has no ICOM code."  Parenthesizing the end where the arrow connects to the box says, "This arrow does not appear in detail diagrams.  Its ICOM code is not tracked from here on and may never be explicitly referenced."  It is possible for an arrow to have a parenthesized arrowhead, disappear for one or more levels of detail, and then be reintroduced at some specific level of detail with a parenthesized end.  If the original source or destination is known, that too should be noted with the appropriate reference expression, written beside the parentheses.
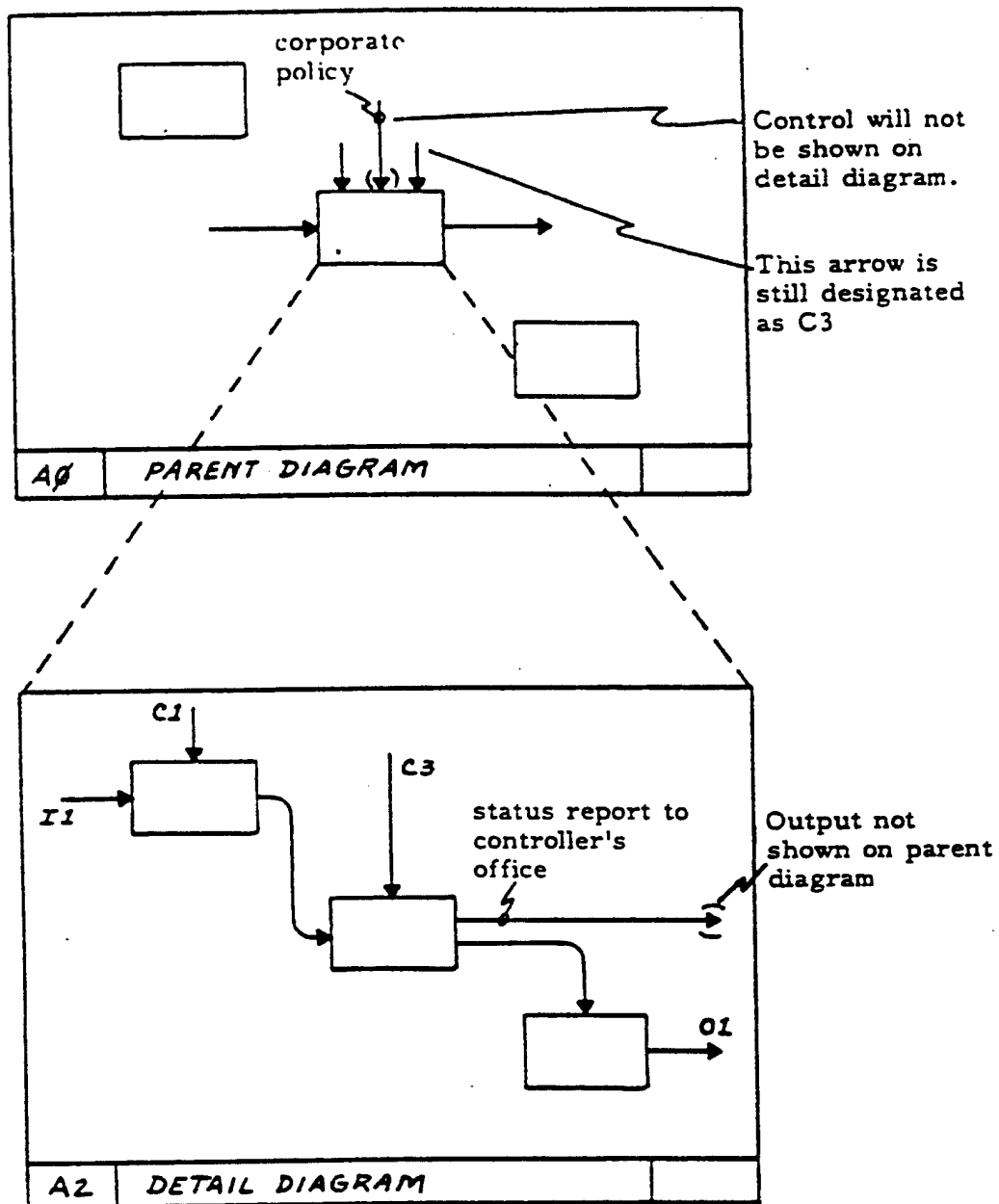
Figure 3-21.   Example of Tunnelled Arrows

### 3.2.5 An Example of Decomposition

Using the Function Model of "Manufacture Product" (MFG0), it is possible to view the overall function of "Manufacture Product" as being composed of the following subfunctions:

- Plan for Manufacture

- Make and Administer Schedules and Budgets

- Plan Production

- Provide Production Resources

- Obtain Manufacturing Materials

- Produce Product

Any of these may then be further subdivided. "Plan for Manufacture" may contain the subfunctions:

- Assume a Structure and Method of Manufacture

- Estimate Requirements, Cost, Time to Produce

- Develop Production Plans

- Develop Support Activities Plans.

Each of these subfunctions may be divided again to the limits of usefulness, knowledge, or time available.

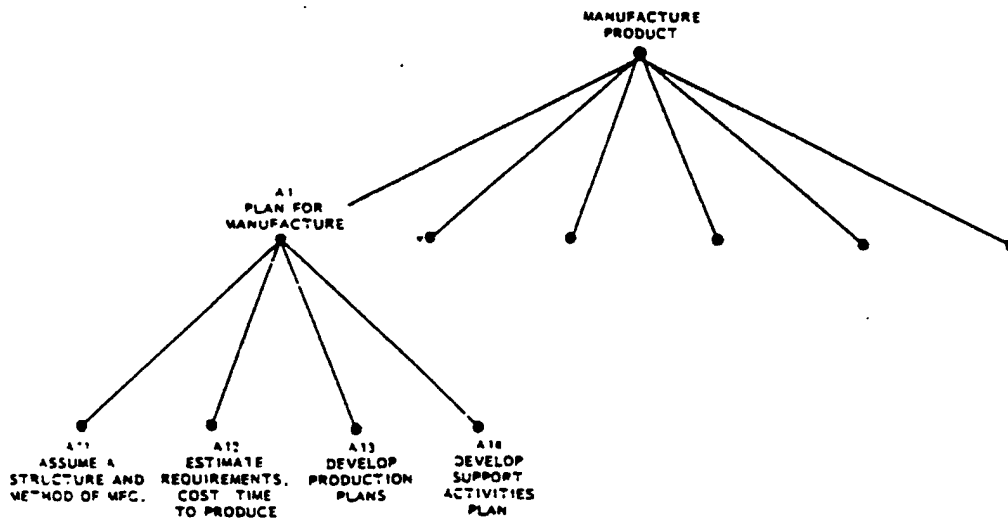This structure of functions and subfunctions may be shown as a "node tree." (Figure 3-22).

**Figure 3-22. Node Tree**

The structure of functions and subfunctions may also be shown as a "node index." (Figure 3-23). This index format is similar to the format of a table of contents and to the format of an "indentured parts list" (bill of materials) used in manufacturing and engineering.

A0    Manufacture Product

A1    Plan for Manufacture

    A11    Assume a Structure and Method of Mfg.

    A12    Estimate Requirements, Cost, Time to Produce

    A13    Develop Production Plans

    A14    Develop Support Activities Plan

A2    ...

**Figure 3-23. Node Index**

The following IDEF$_0$ diagrams begin with the same subject found at the top of the "node tree." Figure 3-24, A-0, shows information and objects that bound what we mean by "Manufacture Product." The A-0 diagram establishes a <u>context</u> for describing "Manufacture Product."
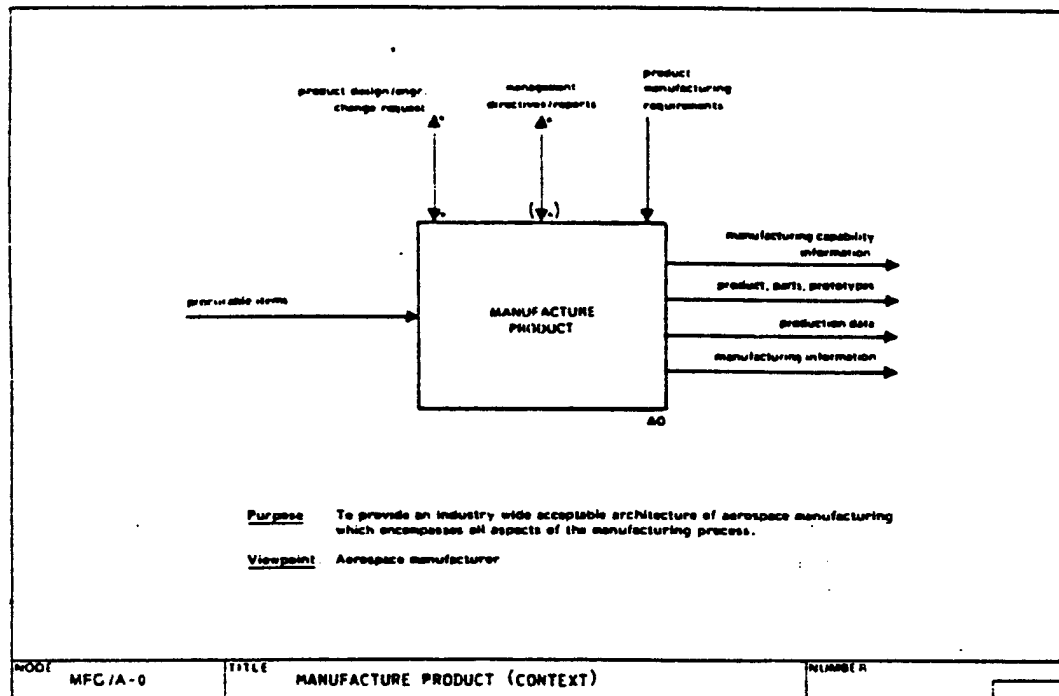


Figure 3-24. A-0, Manufacture Product

"Manufacture Product" includes everything that starts with the Inputs and Controls:

Procurable Items
Product Design
Product Manufacturing Requirements

and finishes with the Outputs:

Manufacturing Capability Information
Product, Parts, Prototypes
Production Data
Manufacturing Information

What happens within the box shown on A-0, "Manufacture Product," is shown on the A0 diagram, which has the same title.
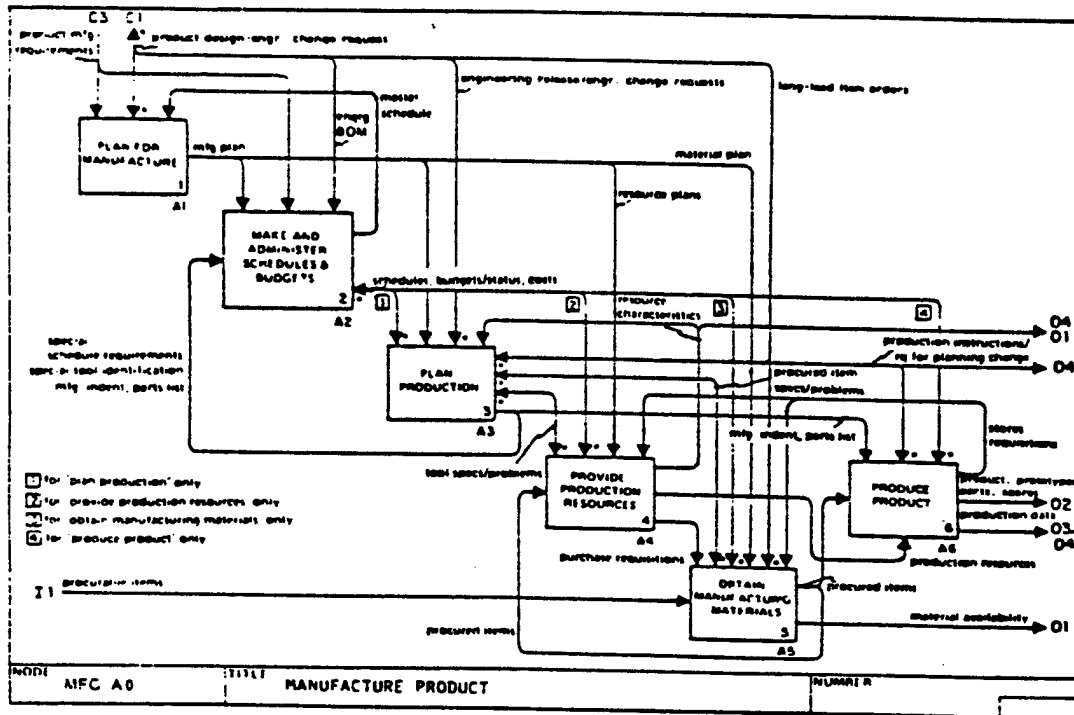


Figure 3-25. A0, Manufacture Product

Each function that is part of "Manufacture Product" is a recognizable grouping of detailed decisions and actions selected from the complex fabric of manufacturing. The function groupings are defined by their data relationships, that is, the information and objects that pass between the functions. These relationships define the boundaries of a function and the terminology used to name each function.

Any of the boxes on A0 may be further described with another diagram. The first box of A0 is detailed in A1 "Plan for Manufacture". A1 shows only the functions and data that are part of "Plan for Manufacture" as defined by the arrows appearing on A0.
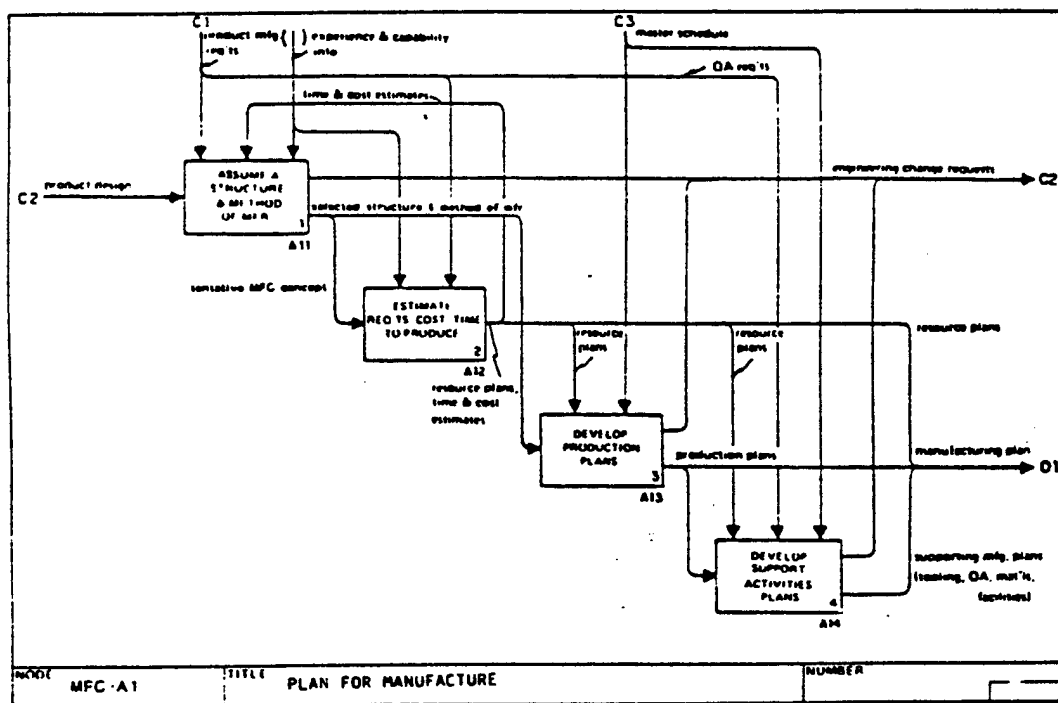


Figure 3-26.  A1, Plan for Manufacture

SECTION 4

READING IDEF$_0$ DIAGRAMS

4.0  Introduction

A model is made up of a series of diagrams and associated materials arranged in hierarchic manner. A node index or table of contents must be provided. Placing the diagrams in hierarchical order gives an overall view of the model and allows access to any portion.

Reading is done top-down. After the top levels are read, first level diagrams are read, then second level diagrams are read, etc. If specific details about a model are needed, the node index is used to descend through the levels to the required detail.

When published, a model is bound in "page-pair" format and "node index" order. "Page-pair" format means that each diagram and the entire text associated with it appear on a pair of facing pages.
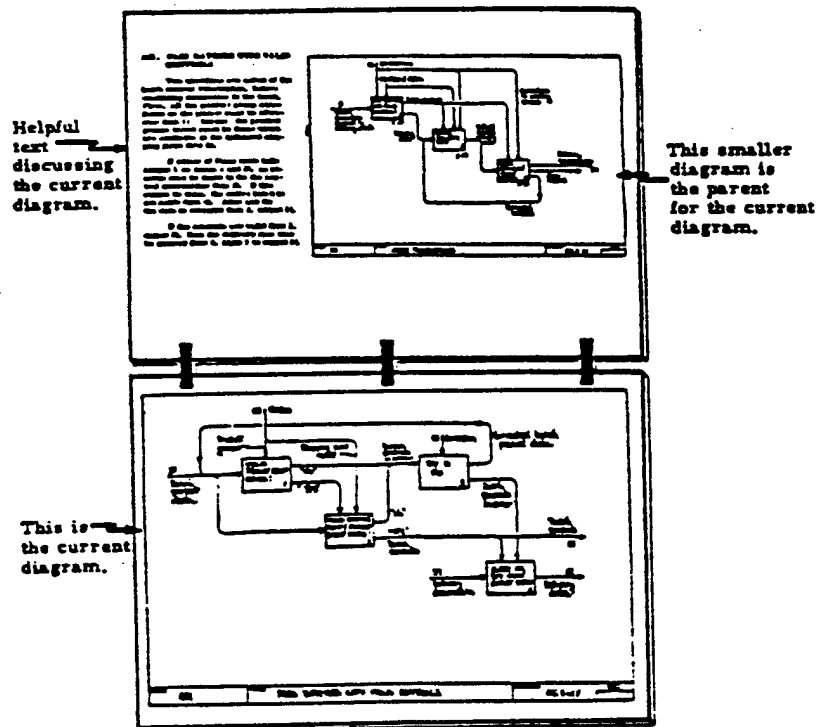


Figure 4-1.  Page-Pair Format

"Node index" order means that <u>all</u> detail diagrams relating to one box on a diagram are presented <u>before</u> the details of the next box. This places related diagrams together in the same order used in an ordinary table of contents.
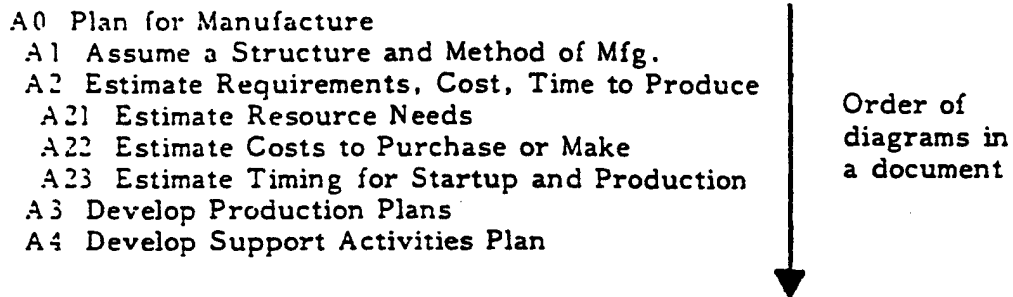
A0  Plan for Manufacture
 A1  Assume a Structure and Method of Mfg.
 A2  Estimate Requirements, Cost, Time to Produce
  A21  Estimate Resource Needs
  A22  Estimate Costs to Purchase or Make
  A23  Estimate Timing for Startup and Production
 A3  Develop Production Plans
 A4  Develop Support Activities Plan

Order of
diagrams in
a document

Figure 4-2.  Node Index Showing Diagram Order

4.1    Approaching a Model

Models provide an overview of the whole system as well as details of a particular subject. To read a model for its overview, use the index to find all high-level diagrams.
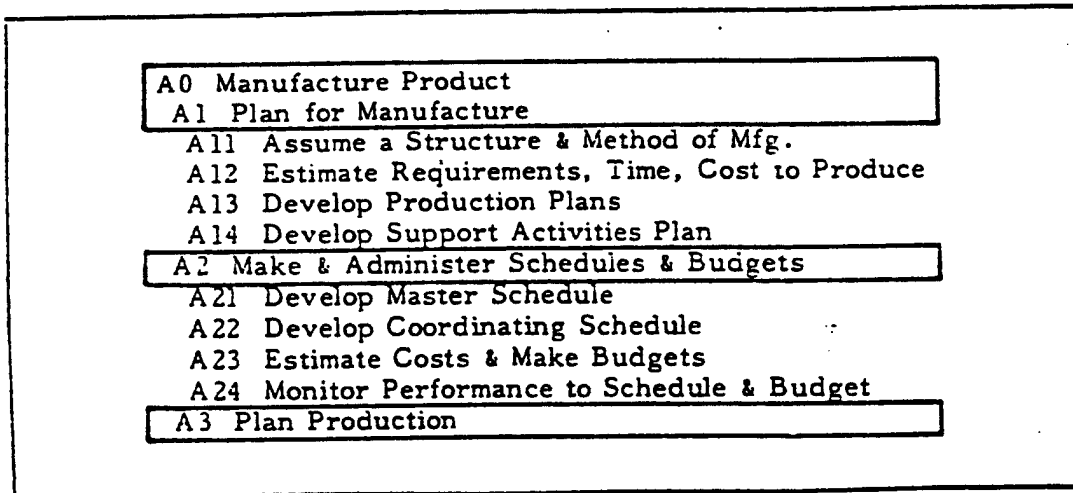
```
┌──────────────────────────────────────────────────────┐
│   ┌──────────────────────────────────────────────┐   │
│   │ A0  Manufacture Product                      │   │
│   │ A1  Plan for Manufacture                     │   │
│   └──────────────────────────────────────────────┘   │
│       A11   Assume a Structure & Method of Mfg.      │
│       A12   Estimate Requirements, Time, Cost to Produce │
│       A13   Develop Production Plans                  │
│       A14   Develop Support Activities Plan           │
│     ┌────────────────────────────────────────────┐   │
│     │ A2  Make & Administer Schedules & Budgets  │   │
│     └────────────────────────────────────────────┘   │
│       A21   Develop Master Schedule                  │
│       A22   Develop Coordinating Schedule             │
│       A23   Estimate Costs & Make Budgets             │
│       A24   Monitor Performance to Schedule & Budget  │
│     ┌────────────────────────────────────────────┐   │
│     │ A3  Plan Production                        │   │
│     └────────────────────────────────────────────┘   │
└──────────────────────────────────────────────────────┘
```

Figure 4-3.  Node Index Showing Overview Diagrams

To read a model for detail, use the index to find all diagrams detailing the subject of interest.

```
┌──────────────────────────────────────────────────────┐
│   A0  Manufacture Product                            │
│    A1  Plan for Manufacture                          │
│      A11   Assume a Structure & Method of Mfg.       │
│      A12   Estimate Requirements, Time, Cost to Produce │
│      A13   Develop Production Plans                   │
│      A14   Develop Support Activities Plan            │
│     ┌────────────────────────────────────────────┐  │
│     │ A2  Make & Administer Schedules & Budgets  │  │
│     │  A21   Develop Master Schedule             │  │
│     │  A22   Develop Coordinating Schedule       │  │
│     │  A23   Estimate Costs & Make Budgets       │  │
│     │  A24   Monitor Performance to Schedule & Budget │  │
│     └────────────────────────────────────────────┘  │
│      A3  Plan Production                             │
└──────────────────────────────────────────────────────┘
```

Figure 4-4.  Node Index Showing Specific Detail Diagrams

Further detailing in a model may be traced by referring to the reference expression just below the box number. This indicates either the node number or page number of the detail diagram for that box. If no reference expression appears, the box has not yet been detailed. For example, on diagram A24 [____] means that the details of box 3 are found on diagrams with node number A243.

Details may be shared within a model or found in a different model. In this case, a down arrow indicates where the shared detailing appears. (See Page 3-10).
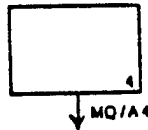
Box 4 is detailed in model MQ by diagram A4. (This is known as a "call.")

## 4.2 Diagram Reading Steps

The precise information about a system is in the diagrams themselves. The following reading sequence is recommended:

1. Scan the boxes of the diagram to gain an impression of what is being described.

2. Refer back to the parent diagram and note the arrow connections to the diagram. Try to identify a "most important" input, control and an output.

3. Consider the arrows of the current diagram. Try to determine if there is a main path linking the "most important" input or control and the "most important" output.

4. Mentally walk through the diagram, from upper left to lower right, using the main path as a guide. Note how other arrows interact with each box. Determine if there are secondary paths. Check the story being told by the diagram by considering how familiar situations are handled.

5. Check to see if a related "FEO" diagram exists.

6. Finally, read the text and glossary if provided.

This sequence ensures that the major features of each diagram receive attention. The text will call attention to anything that the author wishes to emphasize. The glossary will define the author's interpretation of the terminology used.

Each diagram has a central theme, running from the most important incoming boundary arrow to the most important outgoing boundary arrow. This main path through the boxes and arrows outlines the primary function of the diagram. Other parts of the diagram represent qualifying or alternative conditions which are secondary to the main path.

The system's operation can be mentally envisioned by pursuing the main path. Specific kinds of data inputs, the handling of errors, and possible alternative outputs lend detail to the story. This walk-through enhances understanding of the diagrams.
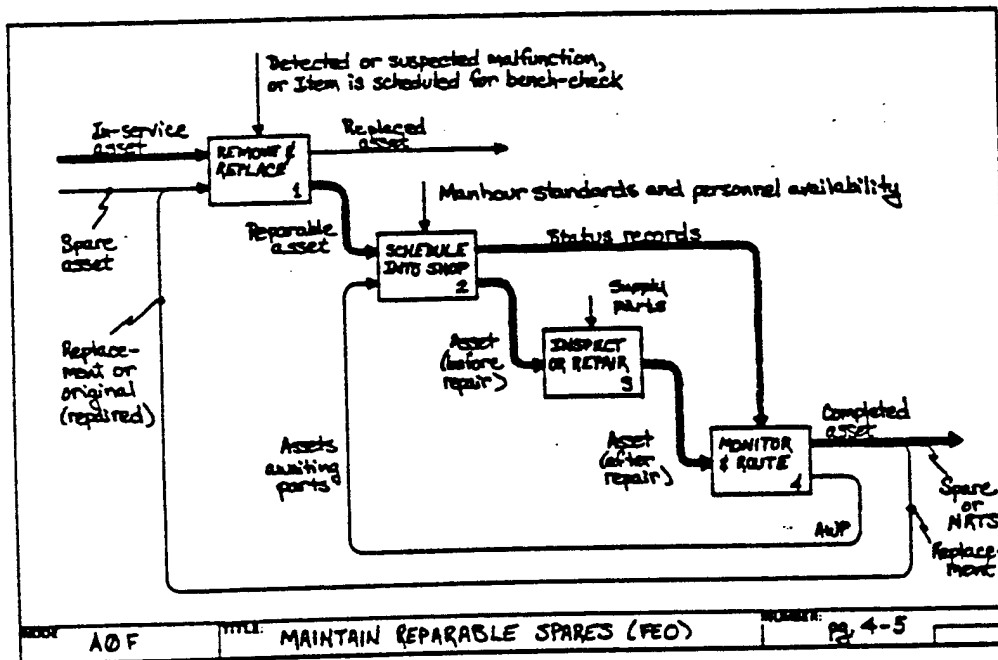


Figure 4-5. Example of Main Path

## 4.3    Semantics of Boxes and Arrows

The fundamental notion which must guide the interpretation of any diagram, or set of diagrams is:

> Only that which is explicitly stated is necessarily implied.

This derives from the very nature of constraint diagrams. Unspecified constraints must not be assumed; necessary constraints must be explicit.

The corollary is that:

> Any further detailing not explicitly prohibited is implicitly allowed.
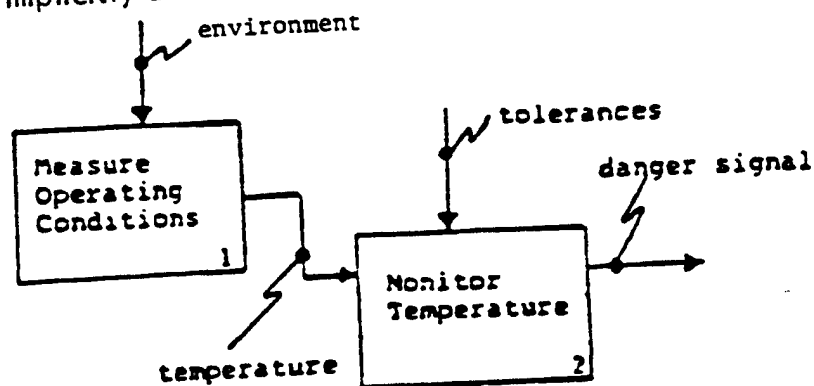


Figure 4-6.  Example of Constraint

An assumption can be made using Figure 4-6 that the temperature is measured "often enough" and the tolerances are changed "when appropriate" and the temperature is monitored against the tolerances "often enough" that the danger signal will be produced "soon enough." None of these intuitive understandings would conflict with subsequent detailing which showed that:

a.    the temperature was measured by periodic sampling, or

b.    current tolerances were requested only when the temperature increased by some fixed amount, or

c.    a series of temperature values produced by box 1
was stored by box 2 which examined the pattern
of change to determine if the pattern was within
the tolerances.

d.    etc., etc.

The graphic notations of a diagram are, by themselves, abstract.
However they do make important fundamental distinctions. Their
abstract nature should not detract from the intended breadth of possible
interpretations that are permitted.

### 4.3.1 Constraints Omit How and When

Either of the two representations:
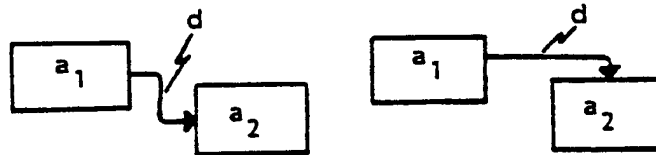


says that:

the activity $a_2$ is dependent on "d"
which is created or modified by the
activity $a_1$.

Each representation defines a constraint relationship between the two
boxes. All that is _explicitly_ stated by the intermediate arrow for
either representation is expressed as follows: some activation of box 2
requires something called "d" that is produced by some activation of box 1.

Frequently, diagrams imply strongly that two or more boxes must
contend for the contents of an arrow. The meaning of the boxes and
arrows shown in Figure 4-7 is that something produced by box 1 is needed
by box 2 and by box 3. It may be that an activation of the arrow's "source"
(box 1) must precede every activation of its "destination" (box 2 or box 3).
It may be that one activation of the source is sufficient for every activation
of any destination. Without additional information, the boxes and arrows
alone permit either interpretation.
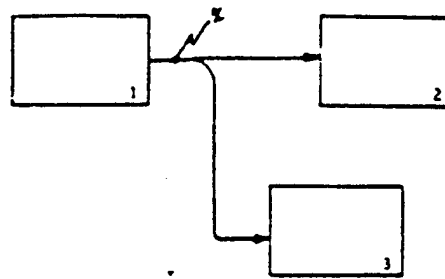
Figure 4-7.  Boxes contending for contents of an arrow

## 4.3.2  Multiple Inputs, Controls, and Outputs

The basic interpretation of the box shown below is:  In order to produce any subset of the outputs [O1, O2, O3], any subset of the entries [I1, I2, I3, C1, C2, C3, C4] may be required.  In the absence of further detailing it cannot be assumed that:

a.      any output can be produced without all entries present, or

b.      any output requires all entries for its production.
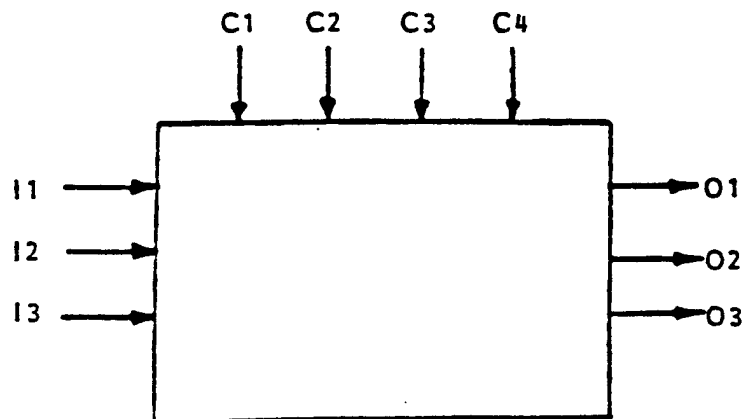


Figure 4-8.  Illustration of Multiple ICOMs

The partial detailing of the previous box (as it might appear in an FEO diagram) indicates that I3, C2, C3, C4 are not required for producing O1. This illustrates the points that:

a. some form of further detailing will specify the exact relationship of inputs and controls to outputs;

b. until that detailing is provided, limiting assumptions about relationships "inside" each box should not be made;

c. reading of a diagram should concentrate on the arrows, which are explicit, rather than on box contents, which are only implicit.
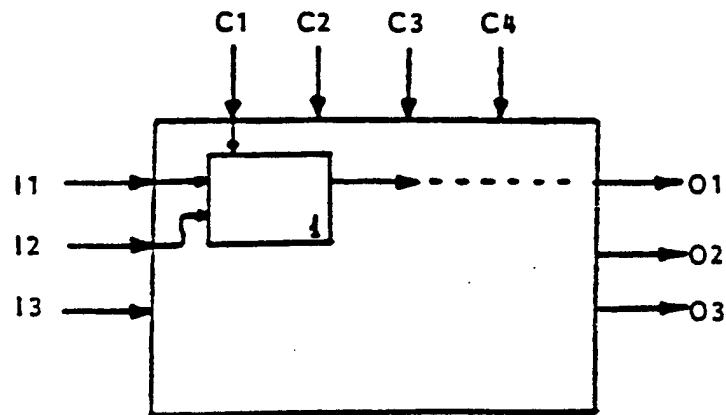


Figure 4-9. FEO representing detailing of multiple ICOMs

SECTION 5

## IDEF KIT CYCLE FORMS AND PROCEDURES

5.1    IDEF Teamwork Discipline

The development of any IDEF model ($IDEF_0$, $IDEF_1$, and $IDEF_2$) is a dynamic process which requires the participation of more than one person. Throughout a project the draft portions of a model are created by authors and distributed to other project members for review. These draft portions of a model are called Kits and may contain diagrams, text, glossary or any other information the author feels is pertinent to the development of the model.

The IDEF teamwork discipline identifies all persons interested in the review of a model as reviewers. Reviewers who are expected to make a written critique of a Kit are called commenters. Reviewers who receive a Kit for information only, are not expected to make written comments and are called readers.

The discipline requires that each person expected to make comments about a Kit shall make them in writing and submit them to the author of the Kit. The author responds to each commenter in writing on the same copy. This cycle continues, encompassing all Kits pertaining to a particular model, until the model is complete and recommended for publication.

The evolution of a model is recorded by disseminating a model (with most recent changes) every 3 months in the form of a Kit which is sent to readers to assist them in maintaining current information about the model.

The end effect of this process for organized teamwork is a high assurance that final IDEF models are valid and are well expressed. The Kits are changed to reflect corrections and valid comments. More detail is added by the creation of more diagrams, text and glossary. More comments are made; more changes are included. The final model represents the agreement of the author and reviewers on a representation of the system being modeled from a given viewpoint and for a given purpose.

## 5.2   The IDEF Kit Cycle

In creating a document, materials written or gathered by an author are distributed to commenters in the form of a Standard Kit. Commenters review the material and write comments about it. The commenters return the Kit to the author who reacts to comments and may use the comments to revise or expand the material. The Kit is returned to the commenter with the reactions from the author. This is known as a Kit Cycle. The steps of the Kit Cycle are as follows:

- The author assembles the material to be reviewed into a Standard Kit*. A cover sheet is completed. Copies of the kit are distributed to each of the commenters, and to the author. The original is filed for reference.

- Within the response time specified, the commenter reads the kit and writes comments directly on the copy. The kit is returned to the author.

- The author responds in writing directly on each commenter's copy. The author may agree with the comment, noting it on his working copy, and incorporating it into the next version of the model. If there is disagreement, the author notes the disagreement on the kit and returns it to the commenter.

- The commenter reads the author's responses and, if satisfied, files the kit. (Commented Kits are always retained by the Commenter.) If the commenter does not agree with the author's responses, a meeting is arranged with the author to resolve differences. If this cannot be done, a list of issues is taken to appropriate authority for decision.

This cycle continues until a document is created which represents the careful consideration of all project members. In addition, a complete history of the process has been retained.

The results of this Kit Cycle are a document to which author and commenters have contributed, and, if necessary, a list of issues that require management action.

Throughout the cycle, a project librarian handles copying, distribution, filing, and transfer of kits between authors and commenters.

---

*Types of IDEF Kits are explained in Section 5.3.
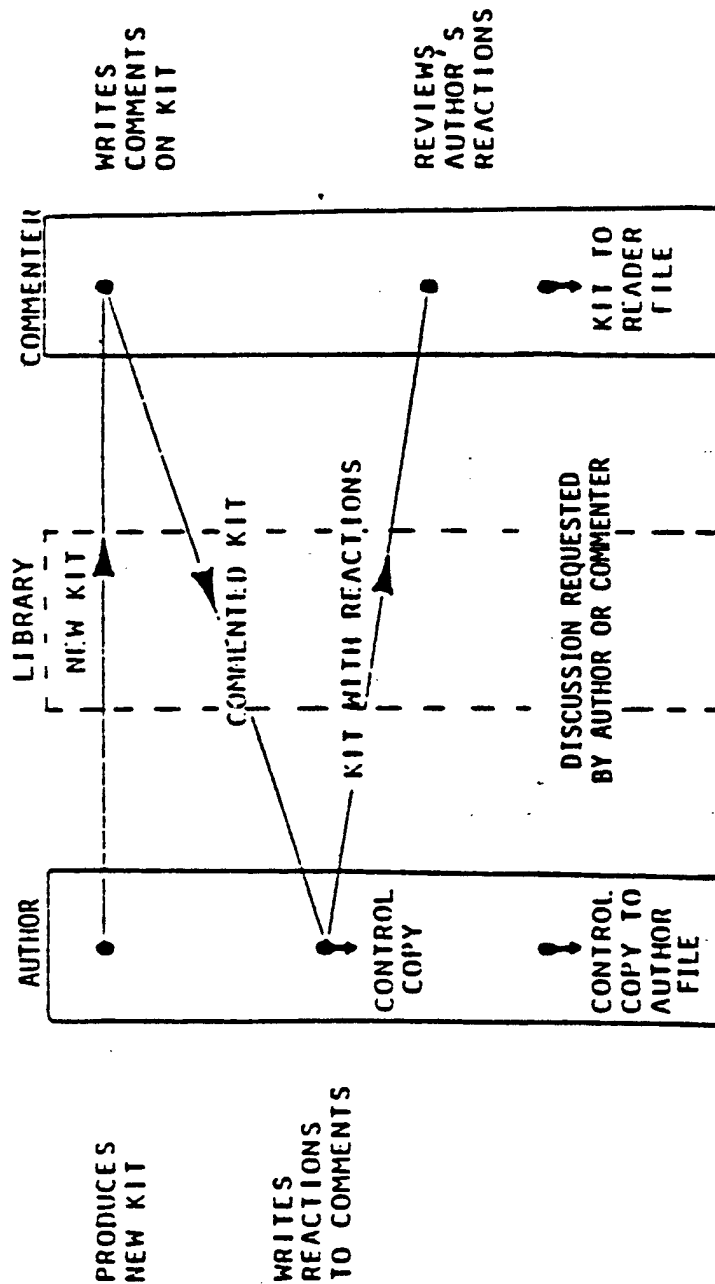
Figure 5-1. Kit Cycle

## 5.2.1  Personnel Roles

The roles and functions of people involved are:

- **Authors (Modelers)** — People who prepare any IDEF model.

- **Commenters (Experts)** — People knowledgeable of the subject being modeled from whom authors may have obtained information by means of interviews, and have enough training in an IDEF technique to offer structured comments in writing.*

- **Readers (Experts)** — People knowledgeable of the subject being modeled from whom authors may have obtained information by means of interviews, and review documents for information but are not expected to make written comments.

- **Librarian** — A person assigned the responsibility of maintaining a file of documents, making copies, distributing kits and keeping records.

A "role" has nothing to do with someone's job title, and the same person may be asked to perform several roles.  Thus, each individual's participation is, in fact, unique and depends upon the kit involved.

## 5.2.1.1  Authors

An author interviews experts and creates documents.  However, an author may or may not be the source of the technical content of a document.  An author may serve only as a technical writer or scribe to record material gathered from other sources.  An author often operates in a role which is largely editorial:  identifying, sorting out, and organizing the presentation of knowledge obtained from experts.

---

*Comments between commenter and author are considered privileged information.  Commented kits are not duplicated for distribution to anyone else on the program.  The library does not retain a file of commented copies.

## 5.2.1.2 Commenter

Commenters read material produced by authors and verify its technical accuracy. Commenters are responsible for finding errors and suggesting improvements. The role of a commenter is the key to producing high quality results. The commenter determines whether the author has followed the IDEF techniques consistently, whether the viewpoint and purpose have been adhered to and whether errors or oversights exist which should be brought to the author's attention.

## 5.2.2 Guidelines for Authors and Commenters

## 5.2.2.1 Commenter Guidelines

No set pattern of questions and rules can be adequate for commenting, since subject matter, style, and technique vary so widely. However, guidelines do exist for improving quality. The major criteria for quality are: Will the document communicate well to its intended audience? Does it accomplish its purpose? Is it factually correct and accurate, given the bounded context? Overall guidelines for commenting are:

- Make notes brief, thorough and specific. As long as the author understands that niceties are dropped for conciseness, this makes for easier communication and less clutter.

- Use the ⓝ notation to identify comments. To write ⓝ-note, check the next number off the NOTES list number the note, circle the number, and connect the note to the appropriate part with a squiggle "⤳." (See Section 5.4 Standard Diagram Form)

- Make constructive criticisms. Try to suggest solutions, not just make negative complaints.

- Take time to gather overall comments. These may be placed on the cover or on a separate sheet. (But don't gather specific points onto this sheet when they belong on the individual pages.) Agenda items for author/commenter meetings may be summarized. Make agenda references specific.

The length of time spent critiquing depends on a variety of things: familiarity with what is being described, the number of times something has been reviewed, the experience of the commenter and author, etc. A kit returned to an author with no comments means that the commenter is in total agreement with the author. The commenter should realize that there is a shared responsibility with the author for the quality of the work.

## 5.2.2.2 Author/Commenter Interchanges

When a commenter returns a kit, the author responds by putting a "√" or "X" by each ⓝ-note. "√" means the author agrees with the commenter and will incorporate the comment into the next version of the kit. "X" means the author disagrees. The author must state why in writing where the comment appears. After the author has responded to all comments, the kit is returned for the commenter to retain.

After reading the author's responses, it is the commenter's responsibility to identify remaining points of disagreement and to request a meeting with the author. This specific list of issues forms the agenda for the meeting.

## 5.2.2.3 Meeting Rules

Until comments and reactions are on paper, commenters and authors are discouraged from conversing.

When a meeting is required, the procedure is as follows:

1. Each meeting should be limited in length.

2. Each session must start with a specific agenda of topics to be considered and must stick to these topics.

3. Each session should terminate when the participants agree that the level of productivity has dropped and individual efforts would be more rewarding.

4.  Each session must end with an agreed list of action
    items which may include the scheduling of follow-up
    sessions with specified agendas.

5.  In each session, a "scribe" should be designated to
    take minutes and note actions, decisions, and topics.

6.  Serious unresolved differences should be handled
    professionally, by documenting both sides of the
    picture.

The result of the meeting should be a written resolution of the
issues or a list of issues to be settled by appropriate managerial decision.
Resolution can take the form of more study by any of the participants.

## 5.3   IDEF Kits

A Kit is a technical document. It may contain diagrams, text,
glossaries, decision summaries, background information, or anything
packaged for review and comment.

An appropriate cover sheet distinguishes the material as a kit.
The cover sheet has fields for author, date, project, document number,
title, status, and notes.

There are two types of IDEF Kits:

- Standard Kit -   All kits to be distributed for comment.
  It is considered a "working paper" to
  assist the author in refining his total
  model and is limited to 20 pages.

- Summary Kit -    Contains the latest version of a model.
  It is sent for information only and is
  designed to aid in maintaining current
  information about the total model while
  portions of the model are being pro-
  cessed through the Kit Cycle.

Standard Kits contain portions of a model and are submitted frequently as work progresses. Standard Kits are submitted through the IDEF Kit Cycle for review and are the type referred to in this manual.

Summary Kits are submitted every three months. These kits contain the latest version of the model. Recipients of Summary Kits are not expected to make comments on them although they may choose to do so. Summary Kits are kept by the recipients for their files. A description of Summary Kits is found in the "ICAM Library User's Guide."

### 5.3.1 Completing a Cover Sheet for a Standard Kit

Complete one cover sheet for each kit submitted. (No reproductions). Fill in the following fields on the Cover Sheet (Figure 5-2).

1. MODEL/DOCUMENT DESCRIPTION:

   Title - Should be descriptive of the kit
   Life Cycle Step - "AS IS" or "TO BE"
   IDEF Method - 0, 1 or 2
   System - Acronym for System or Subsystem
   Distribution Type - Specify if other than Standard
   Kit Distribution*

2. PROJECT INFORMATION:

   Author - Name of person submitting kit**
   Date - Date sent to Library
   Company - Name of company submitting kit
   A.F. Project No. -
   Task No. -

3. KIT INFORMATION:

   Check Standard Kit, indicate document number assigned
   by Library if this is a revision to a Standard Kit

4. REVIEW CYCLE:

   To be signed and dated after review by commenter
   and author.

---

*Types of Distribution available are discussed in Volume XI of the (ICAM) Architecture Part II Final Report AFWAL-TR-81-4023.

**In cases where a Standard Kit is submitted as a group effort (i.e., task team, committee, or co-authors) one individual from the group assumes responsibilities as "author."

Figure 5-2. IDEF Cover Sheet

5.     NODE INDEX/CONTENTS:

Node number, title and C-number of each page
of the document (including the cover sheet)
CONTENTS SHEET, Figure 5-4 (if needed) is
always Page 2.

6.     COMMENTS/SPECIAL INSTRUCTIONS:

Any other information for the reviewers. This
can also be used for special instructions to the
library about the handling of the document. The
library also uses this field for special
instructions to receiver of kits.

## 5.3.2  How to Prepare a Standard Kit

To avoid oversights, review the kit as if that were the only infor-
mation available. Catch any typographical errors. Add points of clari-
fication that come to mind as brief notes on the kit itself. Glossary
definitions for terms that appear in the kit should always be appended
as support material.

Gather helpful materials and append these for the commenter's
benefit. Never use this supplemental material to convey information which
should properly be conveyed by the diagram itself. Whenever possible, use
the most natural means of communication - diagrams - to show details
that are important for the reader in understanding the concepts. Combine
all material with a completed Cover Sheet and Node Index/Contents Sheet
and submit to the Library.

## 5.4  Standard Diagram Form

The Diagram Form (Figure 5-5) has minimum structure and
constraints. The sheet supports only the functions important to the
discipline of structured analysis. They are:

- Establishment of context;

- Cross-referencing between pieces of paper;

- Notes about the content of each sheet.

| Pg. | Node | Title | C-Number | Status |
|-----|------|-------|----------|--------|
| 15 | PI/E115G1 | Department | DSC1095 | |
| 16 | PI/E116G1 | Employee | DSC1096 | |
| 17 | PI/E120G1 | Part Specification | DSC1097 | |
| 18 | PI/E170G1 | Purchase Request | DSC1098 | |
| | | Approval Authority | | |

| Pg. | Node | | | mber | | tus |
|-----|------|---|---|------|---|-----|
| | | | | | | |

| NOMENCLATURE | TITLE | | C-NUMBER | |
|--------------|-------|---|----------|---|
| | NODE INDEX/CONTENTS SHEET | | DS 1116 | PAGE |

STD 1

Figure 5-4. IDEF Node Index/Contents Sheet

The diagram form is a single standard size for ease of filing and copying. The form is divided into three major sections:

- Working Information (top)

- Message Field (center)

- Identification Fields (bottom)

The form is designed so that the working information at the top of the form may be cut off when a final "approved for publication" version is completed. The diagram form should be used for everything written.

## 5.4.1 Working Information

### The "Author/Date/Project" Field

This tells who originally created the diagram, the date that it was first drawn, and the project title under which it was created. The "date" field may contain additional dates, written below the original date. These dates represent revisions to the original sheet. If a sheet is re-released without any change, then no revision date is added.

### The "Notes" Field

This provides a check-off for $n$ notes written on the diagram sheet. As comments are made on a page, the notes are successively crossed out. The crossing out provides a quick check for the number of comments, while the circled number provides a unique reference to the specific comment.

### The "Status" Field

The status classifications provide a ranking of approval. They are:

WORKING              The diagram is a major change, regard-
                     less of the previous status. New dia-
                     grams are, of course, working copy.

DRAFT

The diagram is a minor change from the previous diagram, and has reached some agreed-upon level of acceptance by a set of readers. Draft diagrams are those proposed by a task leader, but not yet accepted by a review meeting of the technical committee or coalition.

RECOMMENDED

Both this diagram and its supporting text have been reviewed and approved by a meeting of the technical committee or coalition, and this diagram is not expected to change.

PUBLICATION

This page may be forwarded as is for final printing and publication.

The "Reader/Date" Field

This area is where a commenter should initial and date each form.

The "Context" Field

This indicates the context in which the diagram form is to be interpreted. The context sketch always is at the next higher level from the current diagram. The current diagram is shown as a box in the sketch, to highlight it; all other parts of the higher level are drawn as ovals. Arrows are omitted. The node number of the higher level diagram is written in the lower left of the Context field.

Node number
of context

CONTEXT:

A32    4

Figure 5-6.   Illustration of Context Field

The Context field of a context diagram (A-0) is "NONE." The Context field of a top level diagram (A0) is "TOP."

#### The "Used At" Field

This is a list of diagrams, other than the immediate context, which use this sheet in some way.

### 5.4.2 The "Message" Field

The Message field contains the primary message to be conveyed. The field is normally used for diagramming. However, the field can be used for any purpose: glossary, checklists, notes, sketches, etc. The author should use no paper other than diagram forms.

### 5.4.3 The "Title" Field

The Title field contains the name of the material presented on the Diagram Form. If the Message field contains a diagram, then the contents of the Title field must precisely match the name written in the parent box.

### 5.4.4 The "Number" Field

This field contains <u>all</u> numbers by which this sheet may be referenced.

#### C-Number

The C-number is composed of two or three letters of the author's initials followed by a number sequentially assigned by the author. This C-number is placed in the lower left corner of the Number field and is the primary means of reference to a sheet. Every diagram form used by an author receives a unique C-number. When a model is published, the C-number may be replaced by a standard sequential page number (e.g., "pg. 17").

#### Page Number

A kit page number is written by the librarian at the right hand side of the Number field. This is composed of the document number followed by a number identifying the sheet within the document.

## 5.5    Keeping Files

Each person participating in a project should maintain files of the dcouments received. The librarian maintains the master and reference files for each kit submitted during the course of a project. A complete explaination of library files is given in the "ICAM Program Library Main-tenance Procedures," Volume XI .of the (ICAM) Architecture Part II Final Report AFWAL-TR-81-4023.
Variations in the filing process may occur based on individual pre-ferences but it is recommended that these files be maintained:

- Standard Kit Files, maintained by authors and commenters

- Summary Kit, maintained by authors, commenters, and readers

- Working Files, maintained by authors

## 5.5.1    Standard Kit File

This file contains the Standard Kits issued or received. A record of kits filed should be maintained and should include any information that allows convenient access to the contents of the kit.

## 5.5.2    Summary Kit File

This file contains the Summary Kits issues or received. A record of these kits should also be maintained.

## 5.5.3    Working File

This file contains all documentation that has not been submitted in a kit. Work in progress and notes should be kept in this file.

## 5.6    The IDEF Model Walk-Through Procedure

In addition to the Kit Cycle, a Walk-Through Procedure has been developed. This procedure may be used when the participants in building a model can be assembled for commenting.

1.    Present the model to be analyzed by using its node index. This is the model's table of contents and gives the reviewer a quick overview of what is to come.

2.    Present a Glossary of Terms. This will allow each reviewer to replace personal meanings of words with those that the presenting team has chosen. The meanings should not be questioned at this point. A change in meaning now would require many changes in the diagrams.

3.    Present each diagram for review.

The diagram walk-through process is an orderly, step-by-step process where questions can be asked that may identify potential weaknesses in the diagram. Six steps of a structured walk-through follow below.

Diagram corrections may be proposed at any step. These corrections may be noted for execution at a later date or adopted immediately.

## Step 1:  SCAN THE DIAGRAM

This step allows the reader to obtain general impressions about the content of the diagram. Typically, the reader will have reviewed the parent diagram which depicted the current diagram as one of its boxes. The reader is now examining how the author decomposed that box.

CRITERIA FOR ACCEPTANCE:

1.    The decomposition is useful for its purpose and complete within the context of its parent box. All lower level functions or data can clearly be categorized under each of its boxes.

2.    The diagram reflects, in the reviewer's opinion, a relevant point of view based on the purpose of the model.

3.  In the opinion of the reviewer, there is enough new information provided to extend understanding of the parent box. There is not so much detail that the diagram appears complex and hard to read.

Unless a problem is rather obvious, criticism may be delayed until Step 3 below. However, first impressions should not be lost. They might be put on a blackboard or flip chart pad until resolved.

Step 2:  LOOK AT THE PARENT

Once the reader understands the current diagram's decomposition, the parent diagram should be reviewed to insure compatibility.

CRITERIA FOR ACCEPTANCE:

1.  The decomposition covers all of the points the reviewer anticipated when reading the parent diagram.

2.  Now that the decomposition of this portion of the parent diagram is revealed, the detail which the reviewer envisioned for this box should still seem correct. If not, note the missing detail.

It might be important at this step to return to the parent diagram briefly and add new (n) - notes or embellish existing ones, based upon the added insight gained from this look at the decomposition.

Step 3:  CONNECT THE PARENT BOX AND THE DETAIL DIAGRAM

This step tests the arrow interface connections from the parent to child.

CRITERIA FOR ACCEPTANCE:

1.  There are no missing or extra interface arrows.

2.  Interface arrows are labeled with proper ICOM codes.

3.  Child arrow labels are the same or an elaboration of its parent's matching arrow. Labels convey the correct and complete arrow contents.

4.     Examination of the connecting arrows reveal no problems in the parent diagram. (An added interface may create a misunderstanding of the message conveyed by the parent.)

A clockwise tour of the four edges of the parent box, checking each arrow, will provide a methodical way to check matching of ICOM codes with parent arrows.

Step 4:   EXAMINE INTERNAL ARROW PATTERN

The pattern of boxes and arrows constitute the primary expression of the model being created.

Each box will be examined in node number order, and each arrow followed in ICOM order for each box. When this process is complete, the reviewers should be led through the diagram to explore the consequences of situations with which reviewers are familiar and to test the diagram's capability to simulate the relationships known to exist.

CRITERIA FOR ACCEPTANCE:

1.     The diagram does not look cluttered. The number of arrow crossings and bends is minimized.

2.     The boxes should be balanced with regard to detail. There should be an equal amount of detail within each box. However, compromises on this criterion are acceptable for the sake of clarity.

3.     The diagram should be consistent with the reviewer's experience and knowledge of the subject matter. Feedback and error conditions should be shown as the reviewer expects.

Step 5:   READ THE SUPPORTIVE DOCUMENTATION

This step examines the points that the author highlights in the text, glossary and FEOs.

CRITERIA FOR ACCEPTANCE:

1.     The text confirms the interpretation obtained from examining the diagram itself.

2.     Normal-paths, feedback, error-handling, and other features suggested by the text are found in the diagram or found in an FEO (For Exposition Only) diagram.

3. Significant diagram features uncovered during Steps 1-4 are found in the text, glossary or FEO.

4. References to the diagram are detailed enough to connect text, glossary or FEO to specific parts of a diagram.

Step 6: <u>SET THE STATUS OF THE DIAGRAM</u>

1. Recommended as it stands.

2. Recommended as modified.

3. Draft: Too many changes made, a redraw is necessary, and future review is required.

4. Not Accepted: A complete re-analysis is required.

SECTION 6

## AUTHOR'S GUIDE TO CREATING IDEF$_0$ DIAGRAMS

### 6.0 Requirements

When creating any IDEF$_0$ diagram, the requirements to be satisfied are that:

    a.    its purpose and viewpoint must match the stated purpose and viewpoint of the overall model;

    b.    its boundary arrows must correspond to those of its parent diagram;

    c.    its content must be exactly everything in its parent box.

### 6.1 Basic Steps of Authoring

The step-by-step discipline of authoring makes it possible to create diagrams that form useful and coherent models. The discipline to follow is:

    a.    <u>Bound the subject matter</u> more precisely than the title of the function box suggests. This is done with a list of data (objects or information) acted on or processed by the function.

    b.    Study the bounded set of subject matter and form <u>possible subfunctions</u> of the total function.

    c.    Look for natural <u>patterns of connection</u> of those subfunctions.

    d.    <u>Split and combine</u> subfunctions to make other boxes.

    e.    <u>Draw a final version</u> of the diagram with careful attention to layout and clarity.

### 6.1.1 Selecting a Context, Viewpoint, and Purpose

Before beginning any model, it is important to determine the model's orientation. This includes the context, viewpoint, and purpose.

The context establishes the subject of the model as part of a larger whole. It creates a boundary with the environment by describing external interfaces.

The viewpoint determines what can be "seen" within the context, and from what "slant." It states the author's position as an observer of or participant in the system for the benefit of an audience. Depending on the audience (management, technical, customer, etc.), different viewpoints may be adopted that emphasize different aspects of the subject.

> Only one viewpoint per model.
>
> One model can serve to reflect only one purpose and can follow only one viewpoint.

The purpose establishes the intent of the model or the goal of communication that it serves. Purpose embodies the reason why the model is created (functional specification, implementation design, customer operations, etc.).

These concepts guide and constrain the creation of a model. While they may be refined as authoring proceeds, they must be consistent throughout a model if its orientation is to remain clear and undistorted.

> Crystalize your purpose.
>
> Without conscious effort, in the process of detailing, the author can stray from the original purpose.
>
> The starting point for every analysis is to bound the context. Decide what the focus is before the top-most box is created. Beware of drifting out of this carefully-selected starting domain.

Every step should be checked against the starting purpose. Things that don't fit may be noted for later modeling of the relevant views.

Clarity is derived from the rigors of detailing. Knowing how far to go, when to stop, when to change gears, and how the pieces fit together will always depend on the purpose for which a model is created.

## 6.1.2  Creating the Context Diagram

To start a model, create the A-0 diagram. Draw a single box containing the name of the function which encompasses the entire scope of the system being described. Use arrows entering and leaving the box to represent the data interfaces of the system to its environment. This single-box diagram bounds the context for the entire model and forms the basis for further decomposition efforts.

Some authors find it easier to sketch the A0 and then draw the single box and interface arrows shown at level A-0. It may be necessary to switch diagramming efforts back and forth between A-0 and A0 several times to obtain a good start for the decomposition.

If the A-0 diagram has begun at too low a level of detail, make the A-0 box the basis of a new level A0 diagram. Move up one level to a new A-0. Repeat this process until an A-0 is reached which has sufficient scope to cover all aspects of the system. (Sometimes such a higher level will broaden rather than clarify the chosen viewpoint. If so, make an A-1 multi-box context diagram and keep the A0 diagram to the original intent.)

### 6.1.3 Creating the Top-most Diagram

All system functions lie within the single box shown on the A-0 diagram. The diagram bounds the context of the system. The A0 diagram decomposes the A-0 diagram into its three to six major subfunctions.

The real "top" of the model is the A0 diagram. It is the first and most important expression of the model's viewpoint. Its structure clearly shows what the A-0 diagram tried to say. The terms and structure of A0 also bound every subsequent level because it is a complete description of the chosen subject. Lower levels delineate each of the A0 functions (boxes). If the purpose of the model is to be achieved, this chain of detailing must be carefully followed at each step. Beginning at the top is the challenge of authoring. It forces the author to maintain a level of abstraction, keep an even model depth, and relegate details to a lower level.

### 6.1.4 Creating Subsequent Diagrams

To form the structure of diagrams, decompose each box on the A0 diagram into its major part. Form a new diagram which covers the same topic as its parent box but in more detail.

To decompose each box into 3-6 boxes, obtain the needed additional facts. Create a first-draft diagram by listing all data items and activities contained in the box being decomposed. Take care that these lists cover the entire topic of the parent box so that no portion is lost in the decomposition. Draw boxes which are based on these lists and draw interface arrows between these boxes.

To derive the clearest possible diagram, modify or re-draw the diagram several times until satisfied. Split (break up a box into two or more parts) and cluster (combine two or more parts into a single box) until satisfied.

Generate portions of more detailed-level diagrams to explore points which need clarification. Create several (3 or 4) diagrams as a set, rather than one diagram at a time.

### 6.1.5  Creating Supporting Material

Eventually, each diagram will be accompanied by a page of narrative text, glossary and, perhaps, FEOs.  The text associated with the A-0 diagram should complete the model's orientation and is written when the A-0 diagram is created.  The text complements the context (expressed in A-0 itself) by expanding upon the stated viewpoint and purpose of the model.

Text for every other diagram (including A0) is quite different. It tells a brief, concise story.  It does not duplicate what the diagram already says, but weaves through its patterns.  At every level, this captures the viewpoint in a way that furthers the purpose.

The glossary explains the definitions the author gives to functions and data in a diagram.  These definitions are important because the terminology used in the model may have a completely different meaning in one company from the meaning in another company.

FEO's (For Exposition Only) are diagrams that highlight a particularly interesting or subtle aspect of a diagram.  They are not bound by IDEF syntax and may contain partial arrow structure, notes, etc. to emphasize their point.

### 6.1.6  Selecting a Box to Decompose

Given a complete parent diagram, "firm up" the higher levels before overcommitting to details.  That is, given A0, emphasize work on A1, A2 A3.  Decomposing A1 into A11, A111, should be done later.  This avoids potential rework should changes be made to higher level diagrams.

Keeping an even depth is not a strict rule.  The amount of depth at any time depends on whether more depth would capture meaning better than one diagram.  Don't put off doing a lower level diagram, e.g. A111, sketch while the ideas are fresh.  The important thing is to treat all such forays as sketches until the "horizontal" even level is confirmed.  Be ready to rework the lower level material if it conflicts with higher level, e.g., A1, A2, A3, (etc.).

Two guidelines are helpful in deciding which box to decompose:

1.   Start with the "hard part" -- the part that is least
     familiar or is least clear.

2.   Select the box whose decomposition will give the most
     information about other boxes.

The simpler topics can be more easily decomposed later, with less risk of error or oversight, and can be easily manipulated to fit the decomposition of the more complex issues.

## 6.1.7  Author Activities

## 6.1.7.1  Data Gathering Phase

### Read Background

The author gathers information about the subject matter by reading source information.

### Interview

The author personally interviews an expert on the subject matter. This interview is not used for multi-person planning or review meetings.

### Think

Digest the information obtained from reading and interviews before actual diagramming begins.

### Pick Box

Decide which box is the appropriate one to detail based on information obtained.

## 6.1.7.2  Structuring Phase

### Draw

This encompasses the actual creative process of generating a diagram. It is not limited only to drawing boxes and arrows. It also includes the listing of random data elements, making sketches, etc., which precede drawing boxes and arrows.

### Redraw

This covers the digestive stage of diagramming and corresponds to editing and rework of verbal text. The activity here is concerned not with creating, but with graphical editing and rearranging for clarity.

### Fix Master

This applies to the correction of master drawings to incorporate improvements. It is primarily a mechanical operation which results from the fact that masters are set aside while changes are made on copies.

## 6.1.7.3  Presentation Phase

### Write and Edit Text

Text accompanying any diagram should be precise. Editing will often alleviate unnecessary detail and redundancy.

### Assemble

Assemble any material, diagrams, node trees, glossaries, text, etc. relevant to the subject. Include a completed Cover Sheet.

## 6.1.7.4  Interaction Phase

### React

This refers to the author reacting to comments. It is a combination of reading and annotating reactions to the comments.

### Talk

This represents time spent when an author and commenter actually get together and talk about author reactions to the comments.

### Group Meetings

This is the time spent in group meetings reviewing progress or brainstorming next steps. The minutes of the group meetings will identify the subject matter under discussion.

## 6.2  Drawing an IDEF$_0$ Diagram

Diagram creation is the most subjective and creative activity of the modeling process.  It is open to wide variations between individual authors.  No one set of steps will work equally well for all authors.  The steps presented here are a proven sequence and are designed to assist a new author in drawing IDEF$_0$ diagrams.

a.    Create a relevant, but not yet structured list of data. List items within the context of the parent box that first come to mind.  Group items, if possible, to show similarities.

b.    Name functions that act on the listed data and draw boxes around the names.

c.    Sketch appropriate arrows.  As each box is drawn, leave arrow "stubs" to make the box more meaningful. Make complete connections as it becomes obvious what the diagram is saying.

d.    Draft a layout that presents the clearest box and arrow arrangement.  Bundle arrows together if the structure is too detailed.  Leave only the essential elements, and modify diagram as necessary.

e.    Create text, glossary and FEO (For Exposition Only) diagrams, if necessary, to highlight aspects which are important.  Propose changes, if needed, in the parent diagram.

## 6.2.1  Generating Function Boxes

Function boxes are generated using the major subfunctions of the parent.  As subfunction names are written, draw boxes around them to form the start of an actual diagram.  At this stage the number of boxes is immaterial.  They can be modified by clustering and splitting.

Clustering will group two or more boxes to form a single box. Its goal is to cluster related functions into a single, more general function. It eliminates premature detail which obscures the message to be conveyed at this level.

Splitting will break a single box into two or more parts. It is the inverse of clustering. Its goal is to provide more detail to present sufficient understanding of the subject being decomposed.

Review the resulting set of function boxes. Look for good balance between the factors chosen. See if the names can be made more specific. Use special terms and abbreviations only when needed to promote communication with the intended audience and only at the detailed diagram levels. Do not use them at the highest (A-0 and A0) levels. Carefully define special terms in the glossary.

In all cases, make the function box names verb phrases. Whenever the phrase can be interpreted as either a verb or a noun, use the notation "(v)" to signify the intended verb usage.

Boxes

1. In most cases, layout boxes diagonally from upper left to lower right. While any layout which makes clear the author's intent is acceptable, vertical or horizontal formats tend to crowd arrows and hinder good structured analysis style.

2. Boxes placed in the upper left "dominate" boxes placed lower and to the right through the control arrows that link them. This standard style makes it easier for readers to understand your meaning.

3. Number each box in its lower right corner. Assign the box numbers on a diagram from left to right and from top to bottom. This defines the node number for each box. The leading digits of the box's complete node number are the same as this diagram's node number. The last digit of the node number is this box number. If the box in Figure 6-1 is in diagram A4, the complete node number for this box is A42.
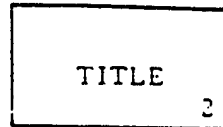
```
┌──────────────┐
│              │
│    TITLE     │
│            2 │
└──────────────┘
```

**Figure 6-1.  Box Number**

4.    On working or draft copies, write the author
      C-number <u>below</u> the lower right corner of any
      box that is decomposed.

5.    No diagram may contain more than six boxes.


## 6.2.2  Creating Interface Arrows

Sketch data interface arrows on each individual box.  Connect ends
of arrows to show which outputs supply which inputs and controls.

<u>Recall that input data are transformed by the function to produce
the output</u>.  If an arrow contains both input and control data. show it as
a control.  If it is uncertain whether an arrow is a control or an input,
make it a control.  If it is unclear whether or not a particular piece of
data is needed at all, leave it out.

Output arrows show the results of possible activations of the
function.  The syntax for output arrows does not indicate which patterns
of output arrows may occur under which circumstances.  If the sequence
is of particular interest, draw a FEO illustrating the pattern.  Do not
worry about sequence.  Just make sure that all important cases are
allowed by the diagram.

Bundle groups of related arrows whenever possible.  The most
common mistake when creating arrows is to make the arrow structure or
the arrow labels too detailed.  The level of detail of arrows must match
the level of detail of boxes.  At high levels, both box names and arrow
labels will be general.

As a final check, compare all arrows to the data list to insure that
each correct data item appears.  Elements that do not appear are either
incorrect for this level of detail or were overlooked when creating the arrows.

### Think control and constraint, not flow

A basic rule for layout of the arrow structure is "constrain, don't sequence." That is, make the diagram structure show relationships that must be true no matter which sequence is followed.

Even though something must progress from stage to stage to reach some desired end result, try to express the constraints that must be satisfied or the invariant properties that must be true rather than some one specific sequence of steps that will yield that result.

The reason is that all boxes may be active simultaneously. Thus, sequence has no meaning.

It is always more powerful to constrain than to sequence. Wherever possible, diagrams should be created that say the right thing regardless of what steps are taken first. Clearly, this is better than restricting to only one of the possible sequences.

Often, it is easiest at first to think of submodule actions in a particular sequence to get unstuck and get something on paper. This may be a good way to get going, but always rework the first attempt into a contraint structure.

### Label carefully

Subordination of unneeded details highlights the meaningful ones.

Don't clutter your diagrams with too much information and too many arrows.

Don't spend too long on a single level. Everything need not be expressed at once to avoid being incomplete and misunderstood.

The whole point of the discipline is to get everything said, eventually.

Also, if there is too much in a diagram, it becomes rigid. If this is allowed to happen, the diagram is weakened. Strength comes from the structure. This can be accomplished by leaving details to the subfunctions.

Do iterate between high-level diagrams and subfunctions that fill out the details.

### Leave out questionable arrows

It is often hard to determine whether to show an arrow or not.

The easiest way to handle the arrow question, is "When in doubt, leave it out." If the arrow isn't really essential to the main backbone, if there are questions about it, it is probably incorrect to include it.

Incorrect omission of a questionable arrow now won't cause permanent damage. The need for the arrow will be clearer in considering subfunctions and the ICOM discipline will force the arrow back up to this level. At that time, there will be no question about it.

### 6.2.3 Level of Effort

The initial goal in generating a diagram should be a clear diagram that represents a definite message and does not violate any syntax rules. When the diagram is finished, the critical guidelines of reading and review by others can be used to improve the first try. Most diagrams can be modified to make a second version that is in some sense better than the first. The first will rarely be the very best.

As skill levels develop, first diagrams get better and authors feel more comfortable using $IDEF_0$. Reworking of diagrams will always be a necessary part of the process. The key idea is to use a review cycle to make progress on paper. In a series of orderly advances, all of the important aspects will be properly handled.

$IDEF_0$ is a thought-forming methodology, not just a diagram-making exercise. Putting thoughts on paper and letting the notation and discipline work is a move towards a satisfactory resolution. Rely on an ability to ask good questions, rather than on the expectation of providing "perfect" answers.

## 6.3    Redrawing an $IDEF_0$ Diagram

### 6.3.1   Modifying Boxes

When first creating a diagram, 3-6 function boxes of approximately the same level of detail are derived. Clustering and splitting will provide a boundary which is more easily understood or which will provide a simpler interaction between the function boxes.

Most often, clustering and splitting work together. Boxes are split and the resulting pieces clustered into new boxes which more closely convey the intended message. The same subject matter is covered, but pieces are grouped in a more understandable way.

### Split and rephrase

It is important that all of the boxes of a diagram have a consistent flavor. Changes elsewhere must not make one subfunction seem out of place. Split and rephrase to restore the balance.

Sometimes a box does not flow with the other boxes of the current diagram.

Frequently the trouble is that other aspects have undergone change and clarification. That which earlier was a good idea, now has the wrong slant or flavor.

Divide the offending box by splitting it into two or more pieces, one of which still contains the essence of the original idea.

Do expect to change the wording of the box (or boxes). With the separation, new ideas become clearer and mesh more closely with related boxes.

## Cluster and replace

A solid abstraction is both clearer and more powerful than premature detail. Cluster related boxes and replace by a single encompassing box.

Frequently a good level of abstraction can be made even better by clustering several boxes into a more general view and postponing the details to the next level down. Draw a line around the cluster and replace them all with one box, suitably named. The extra level is not an added complexity. It is a better representation because the structure has been shown more clearly.

This phenomenon arises often in conjunction with splitting and is one of the most powerful methods of explaining functions.

### 6.3.2 Bundling Arrows

Both arrows and boxes should be at a corresponding level of abstraction in a diagram.

There are two basic ways to achieve this:

1.  Bundle arrows with the same source and destination under a single more general label, and make one arrow.

2.  Rename some boxes (using Split and Cluster) to better distribute the subfunctions and relabel resulting arrows.

It is seldom true that an excess of arrows indicates a mistake. It may be that they are both accurate and precise. But it is always true that an excess of is bad if things are obscured. The ability of readers to understand what is being said should guide the number of arrows used.

### 6.3.3  Proposing Modifications to the Context

The detailed understanding revealed by creating a new diagram may uncover errors or oversights in the parent diagram.  Parent modification is a natural and anticipated event.  When creating the arrow structure, the rule is that "if there is any doubt whether a data arrow is needed by a function box, leave it out -- later detailing will demonstrate whether or not the arrow is really needed."  This is the point where such questions are resolved for a specific reason rather than through former speculation.

Parent diagram changes may represent various degrees of difficulty.  If the change can be accommodated by a revision to the immediate parent only, this is simpler than a revision which involves more remote diagrams as well.  When proposing a change, think it through carefully and assess its complexity.  Substituting simple changes for major ones can harm the quality of the decomposition.  When the correction is completed, check all boundary connections to insure that ICOM codes are properly shown.  Inform other authors working on the diagram of the changes.

Always have in mind the parent diagram for the box being detailed.  It will aid in the creation process.

If the detailed diagram doesn't fit the context, either the current work or the context is wrong.  Change the context or change the current work.  They must match.

### 6.3.4 ICOM Syntax for Connecting Diagrams

An important aspect of understanding diagrams is the ability to find and understand facts that are needed. Node numbers show the structure of the box decomposition. The arrow network provides interface connections.

ICOM codes are written on all arrows having one end unconnected on the diagram. These <u>boundary arrows</u> connect the arrow network across diagrams. Each boundary arrow is labeled with an ICOM code to specify the connection of the arrow to the parent diagram.

### 6.4 Graphic Layout

Layout the boxes diagonally according to the constraint structure, from upper left to lower right, with feedback arrows going up and left. At this point, number the boxes from left to right.

It is best to start this layout with the most heavily used constraint arrows only, adding less used paths later. This subset of the arrows will permit the box position to be determined. Draw all boundary arrows shown on the parent diagram and then add the remaining arrows.

### 6.4.1 Constraints on the Diagram

1. When an entry arrow serves both control and input functions, show it as control. When in doubt, make it control. An arrow appearing on a parent diagram as control can appear on the next level as control or input or both, depending on its relationship to the subfunction at this level.

2. <u>Function boxes must always have control arrows</u>, though they may not always have inputs.

3. In general, do not split an arrow into both a control and an input to the same box. This detail is best shown on a lower level diagram where the destination of each branch and the reason for the split will appear. When it must be done choose labels for the two branches that will convey your important decision.
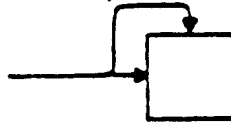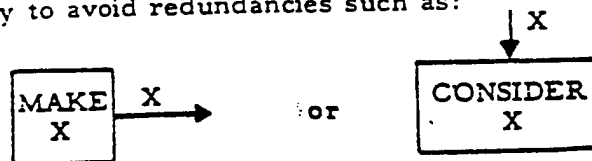
4. Cyclic processing or data storage on an $IDEF_0$ diagram may be shown as

or
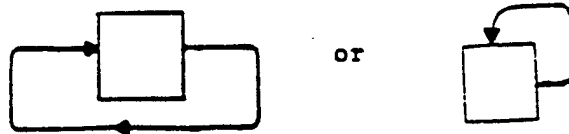
Only show it this way if the storage is to be thought of as being "at this level." Otherwise, show the feedback loop at the next level of detail, that is, "inside" the box.

5. Try to avoid redundancies such as:

MAKE X

or

CONSIDER X

In these cases, the trivial box names merely repeat the message conveyed by placement of the arrows. A little additional thought will usually produce more informative box names.

## 6.4.2 Arrow Placement

1. Draw arrows along horizontal and vertical lines, not diagonally or as curves (except at corners).

2. Place arrow corners, intersections, and labels a reasonable distance away from boxes.

3. Don't use the keywords, i.e., "data," "function," "input," "output," "control," or "mechanism" in names or labels, unless absolutely necessary.

4. If an arrow is long, label it twice.

cook        cook
O2

5. Place ICOM codes at the unconnected ends of arrows.

6. Connect open-ended boundary arrows to show all the places affected. Readers may miss connections otherwise.

A        A
rather than
A
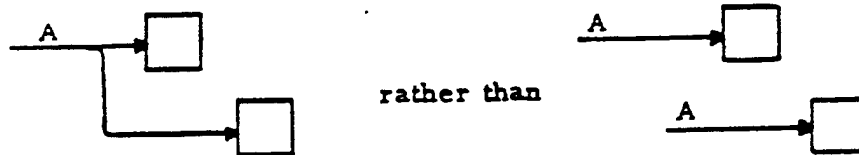
7. Don't draw arrows all the way to the margins of the diagram sheet.

rather than
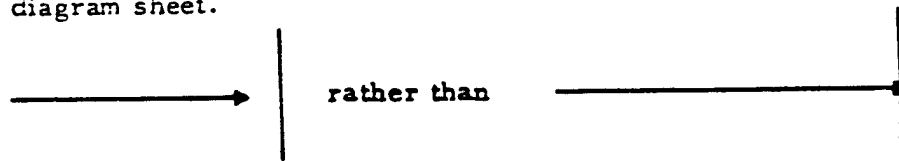
8. Space parallel arrows adequately. They are hard to follow visually if they are lengthy and close together.

rather than
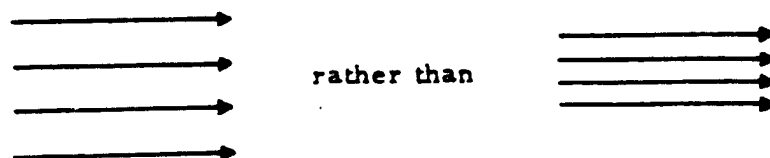
9. Place extra arrowheads along arrows where needed for clarity.

### 6.4.3 Arrow Layout

1.  Bundle arrows with the same source and the same destina-
    tion unless the arrow is of such importance that making
    it part of a pipeline would decrease clarity.

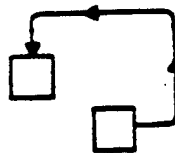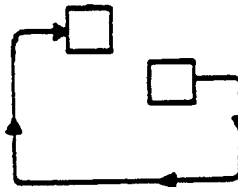    **rather than**
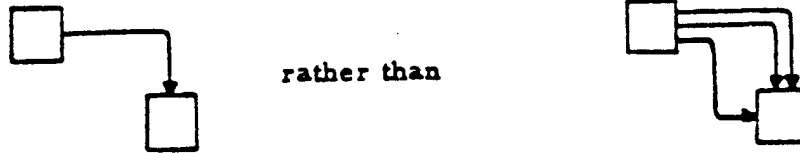
2.  On any one side of a box, there should never be more
    than <u>four</u> arrows. If there are more, bundle some,
    label <u>with</u> a suitable abstract label, and fan out branches
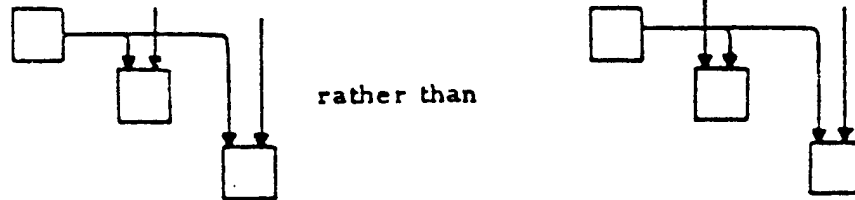    to their destinations.

    **rather than**
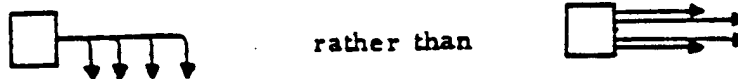
3.  Control feedbacks are clearer when shown as "up and
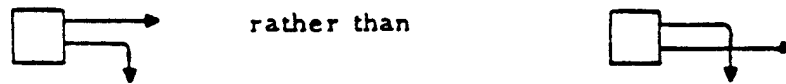    over."

    Input feedbacks are clearer when shown as
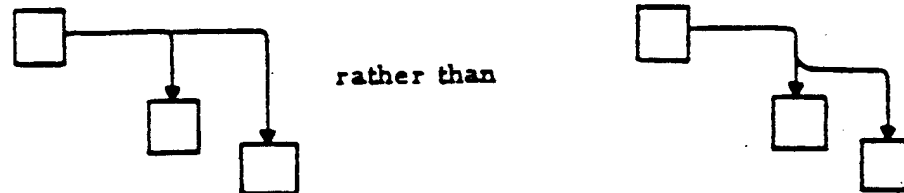    "down and under."

4.  If an arrow branches and feeds into several boxes, draw it at the same relative ICOM position on each box, if possible.
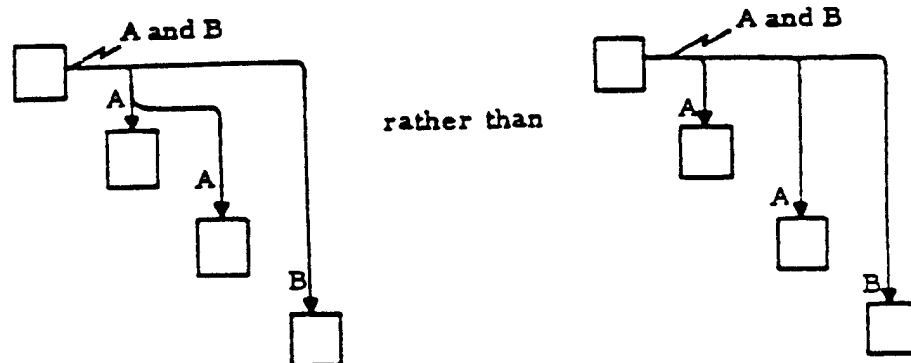


rather than

5.  Lay out arrows so as to minimize unconnected crossings.



rather than

6.  Minimize curves and corners, whenever possible:



rather than

7.  Use the expressive potential of branching arrows when and if it is appropriate:



rather than

8.    To avoid clutter when showing an external arrow which applies identically to or is obtained identically from each and <u>every</u> box on a diagram, use the "to all" convention:
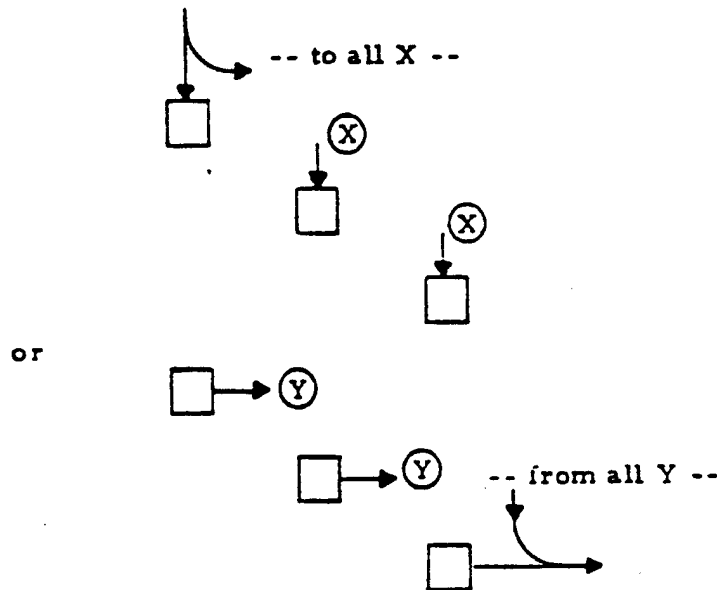


## 6.5   <u>Writing Text</u>

The text that accompanies each diagram presents a brief overview of the diagram. It is always less than a page in length. It highlights features that the author feels are of special interest or significance by walking the reader through the main ideas of the diagram. It does not duplicate every detail shown on the diagram itself. If the diagram conveys the intended message sufficiently well by itself, the text may be omitted.

Write the text only after a diagram has received a fairly high level of review and approval. Waiting to write the text forces the diagram itself to properly communicate the intended message. Text based on carefully drawn diagrams will be as structured and as organized as the diagram itself.

Include glossary definitions to explain the use of terms used in the diagram. A word may have different connotations from company to company and even within one company. For example, "directives" could emanate from the design or finance departments, government, corporate policy, etc.

Try to get good text without adding a FEO. FEO's should be used to illustrate subtle aspects that clarify the intent of a diagram but which would clutter the diagram were they included. If a FEO is necessary, the text that accompanies it should refer to the related diagram.

## 6.5.1 Writing the A-0 Text

Write the A-0 level text when A-0 diagram is drawn and include it whenever A-0 is presented. Since all decomposition proceeds from the A-0 diagram, place any noteworthy facts which apply to the entire model in the text associated with this diagram. The A-0 text must contain a discussion of the viewpoint and purpose of the model.

## 6.5.2 References and Notes

A brief reference language may be used in writing text and notes to refer to diagrams and specific parts of diagrams. For example:

| | | |
|------|-------|------------------------------------------|
| O2 | means | The boundary arrow with ICOM code O2 |
| 2I1 | means | Box 2 Input 1 |
| 2O2 to 3C1 | means | The arrow from 2O2 to 3C1 |
| (n) | means | Note n |

These items may be used individually if they refer to the current diagram (e.g., in Ⓝ notes or text). Otherwise they should be preceded by node number, and if necessary, by model name. A period "." is used to mean "see" a certain thing on the specified diagram. For example:

| | |
|---|---|
| MODEL/A21.3C2 means | In "Model" on diagram A21, see Box 3 Control 2 |
| A42. ③        means | On diagram A42 in this model, see note 3 |

Each reference needs only as much as necessary to be completely unambiguous. The fullest form is:

ACCT/A21(T56).1O2 to 4C3

which means in model "ACCT," diagram A21, version T56, "see" the arrow from Box 1 Output 2 to Box 4 Control 3.

### Add neat and complete references to your text, glossary and FEO

It is easy to be exact without causing the reader to falter.

When writing texts, scan the diagram continually to make the story interesting.

Don't worry about adding references. That will impede the flow and make sentences awkward.

When finished with the text go through and add box-number references to tie the text to the diagram exactly.

Most of the time a simple box number ("Box 3") or a reference to a couple of arrows are sufficient ("Box 3, O1 and C2"). When a reader needs to actually "See" the other diagram, use the "." of the reference language.

References should be used for clarification. They will be unobtrusive only if added after the story-line text is finished.

## 6.6 Model Quality Checklist

The material presented in the following sections provides definable checks for constructing or evaluating the quality of $IDEF_0$ models.

### 6.6.1 Syntax

Syntactic rules are constraints in the sense that they describe logically what is represented graphically. Components of a diagram in $IDEF_0$ can be considered to be primitives; syntactic constraints are statements which specify allowable or valid relations and operations which affect these primitives. In short, a syntactic constraint can be thought of as a criterion: if the constraint is not satisfied, or is violated in the graphic display, the criterion has not been met, and the resulting diagram or model is deficient with respect to that particular constraint.

At least three different types of syntactic constraints are distinguished, depending on the level of analysis or level of graphic representation that is being addressed:

1) Local Construction:

These rules pertain to simple, first-order combinations or associations of primitives. They are defined for FEO's and diagrams, and must be followed for valid construction. Diagrams having missing components or showing relations among primitives other than those stated below are "invalid syntactically."

2) Global Construction:

These rules pertain to whole diagrams, but not to text or FEO's. Thus, they are defined over several constructions, and can be verified only after the diagram has been completed.

3) Model Construction:

Constraints in this category specify the layout of nodes and numbering/lettering designations that create an ordered hierarchy among diagrams.

The following checklists can be used as guidelines for determining a minimum level of acceptability for IDEF$_0$ diagrams and models. Deficiencies can be pinpointed at the local, global or whole model level. "Completeness" and "correctness" are identified here as overall evaluation standards that must be met. Working definitions (for IDEF$_0$ syntax) appear below:

## COMPLETENESS

| DEFINITION | DEGREE TO WHICH ALL REQUIRED FIELD-ENTRIES, LABELS, BOXES, ARROWS, AND IDENTIFYING NOTATIONS ARE PRESENT IN A DIAGRAM OR MODEL. |
|---|---|

## CORRECTNESS

| DEFINITION | DEGREE TO WHICH A DIAGRAM OR MODEL HAS BEEN ACCURATELY CONSTRUCTED, LABELLED, AND IDENTIFIED; I.E., DEGREE TO WHICH SYNTACTIC RELATIONS HAVE BEEN REPRESENTED APPROPRIATELY IN THE GRAPHICS, AND CONSTRAINTS OBEYED. |
|---|---|

| MEASURES | CHECK FOR ADHERENCE TO CONSTRAINTS ITEM BY ITEM. (SEE SECTIONS 6.6.1.1, 6.6.1.2, and 6.6.1.3). |
|---|---|

### 6.6.1.1  Local Construction Syntax

    a.    One way arrow segments are made up of ordered pairs of endpoints (SOURCE, SINK), where:

        SOURCE points are one of:

            boundary endpoint near diagram I, C, or M boundary
            box endpoint on Box O side
            fork sink
            join sink

SINK points are one of:

> boundary endpoint near diagram O boundary
> box endpoint on Box I, C, M side
> join source
> fork source

Note that all combinations are legal except a (diagram boundary, diagram boundary) arrow.

b. <u>Arrow labels</u> are associated with arrow segments. Line squiggles may be used to clarify the association.

c. Each <u>ICOM</u> code from the parent box must be associated with the boundary endpoint of some arrow.

d. Boxes can have an integer number. Box numbers start at 1 and increase by 1.

e. <u>Names</u> are associated with boxes and cannot stand alone.

f. <u>Endpoints or arrows</u> at <u>diagram boundaries</u> have tunnel "( )" or an ICOM but not both.

g. <u>Endpoints of arrows</u> at <u>box endpoints</u> can have tunnels "( )".

## 6.6.1.2 <u>Global Construction Syntax</u>

- A finished <u>IDEF diagram</u> consists of no less than three and no more than six boxes, unless it is an A-0 diagram in which case it contains exactly one large box.

- <u>Each box must be named</u> and numbered. On each diagram the numbers start at one and increase monotonically by one. The box on an A-0 diagram is numbered 0. No two boxes on one diagram can have the same number.

- <u>Each box must have at least one arrow source</u> at its O (right) side and one arrow sink at its C (top) side.

- <u>All endpoints</u> of arrow segments at diagram boundaries <u>must have either an ICOM</u> code or tunnel "( )", except on an A-0 diagram for which there are neither ICOM codes nor tunnel "( )".

## 6.6.1.3 Model Construction Syntax

### Rules and Conventions for Valid Model Structure in IDEF$_0$

A model is a hierarchically structured collection of nodes representing successive refinements, each of which may be detailed on one or more diagrams. The top node of the IDEF$_0$ diagrams in a model is designated AO. A box has a node number derived by appending the box number to the node number of the diagram on which the box appears. A diagram has the same node number as (is the same node as) the box it details. However, when appending the box number, the "0" (from the A0 node) is treated as "null" and dropped. For example, the decomposition of Box 2 on the diagram detailing node A0 is designated A2. Also the decomposition of Box 1 on the diagram detailing node A2315 is designated A23151.

A FEO that illuminates an IDEF diagram is designated by concatenating the letter F to the right-hand end of the designator of the diagram. Text or glossary that illuminates an IDEF diagram or FEO is designated by concatenating the letter T or G, respectively, to the right-hand end of the designator of the diagram or FEO. For example, the FEO illuminating Box 3 of node A21 would be designated A213F. Text illuminating node A114 would be designated A114T. Text illuminating FEO A31F would be designated A31FT. Glossary definitions for terms used on diagram A114 would be designated A114G, etc. Additionally, if there is more than one illuminating FEO or glossary page, a sequence number is concatenated to the right of the letter F, or G. Thus, the two FEO's illuminating diagram A213 would be designated A213F1 and A213F2.

Although the top node of an IDEF function model is always A0, there are other nodes at a "higher" level used to show context and environment. The A-0 (A minus zero) shows the context of A0. The A-0 box may be shown in an A-1 diagram, etc.

The interconnections between parent and child arrows are maintained through the use of ICOM codes. The ICOM occurrences are represented explicitly on the child diagram at the diagram boundary point of an arrow. The corresponding ICOM on the parent box is determined implicitly from the position of the arrow's context point. The letter (I, C, O, or M) corresponds to the side of the box on which the context point occurs. This is followed by a number determined by the order in which the arrow appears on that side of the box (counting top down and left to right). Note that ICOM codes will change as arrows are added, deleted, or moved on the parent box.

## 6.6.2 Semantics

"Semantics" is a term that has been applied to a wide range of factors, all of which have some effect on the "meaning" of labels, diagrams, and models in $IDEF_0$. In order to establish criteria useful for evaluating models, those aspects of "meaning" that relate to the consistent and unambiguous naming of data and function in $IDEF_0$ will be dealt with. Criteria for evaluating the overall "story," "message," or "sense" of a diagram or model will be addressed in terms of accessing complexity.

At least five types of semantics should be evaluated. There are:

1. COMPLETENESS

2. CONCISENESS

3. CONSISTENCY

4. CORRECTNESS

5. COMPLEXITY/UNDERSTANDABILITY

Each of these criteria is considered separately in the summary sections that follow.

## Completeness

One useful measure for evaluating the adequacy of information-coverage is <u>amplification</u>. This refers to the degree to which additional details are introduced in a child diagram, relative to its parent. Ideally, there should be sufficient detailing introduced to make the parent more comprehensive in information content -- but not so much detail that the overall model becomes complex and harder to understand. Also, there should be a <u>balance</u> between the degree of detail conveyed by the function-box labels, and that expressed by data arrow labels. A diagram with very general and abstractly-labelled function boxes, but with minutely detailed labels on the data arrows is <u>unbalanced</u> semantically; the amplification factor might also be low on successive child diagrams because there would be limited room to do further detailing on data arrow labels.

### Definitions and Measures

| | |
|---|---|
| DEFINITION | SUFFICIENCY OF INFORMATION <u>CONTENT</u> TO COVER SUBJECT MATTER (<u>CONTEXT</u>) |
| MEASURES AND PARAMETERS | <u>DIAGRAM LEVEL</u>: FOR ANY DIAGRAM, EXCEPT THE TOPMOST - DEGREE OF ADEQUATE COVERAGE OF INFORMATION WITH RESPECT TO THE BOUNDED CONTEXT ON THE PARENT. |

(PARENT)

DIAGRAM BEING ANALYZED
OR CONSTRUCTED
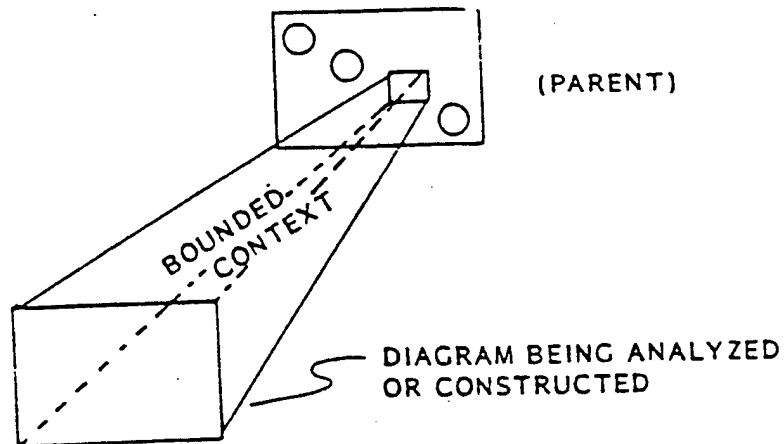
WHOLE MODEL LEVEL:

DEGREE OF DECOMPOSITION OF <u>BOXES</u>.
[VARIES ACCORDING TO <u>PURPOSE</u> OF MODEL;
AMOUNT OF DETAILING NEEDED TO FULLY COMMUNICATE THE PURPOSE]

DEGREE OF DETAILING OF <u>ARROWS</u> FROM TOP TO LOWEST
LEVEL IN A MODEL
[VARIES WITH LEVEL OF DETAIL AND ABSTRACTION OF
FUNCTION BOXES]

## Conciseness

The "information value" of labels and titles in $IDEF_0$ is ultimately
the potential relevance of the diagram content to the reader. For this
reason, authors should strive to select label names that are in common
use, especially if they are associated with technical processes or practices
in the manufacturing environment. The words used on diagrams and in
accompanying texts should also be <u>natural</u>, in the way they relate to
the anticipated reader-audience. For example, an audience of accountants
would relate to labels such as "indirect labor," "direct labor," and
"G&A overhead," but project managers may prefer terms such as
"billable labor," "internally funded research," and "support labor."

"Degree of redundancy" is difficult to measure when simply reading over a diagram or series of diagrams. If authors provide a glossary of key terms, and provide sufficiently detailed listings of characteristics or properties of technical terms, readers and reviewers would have a more objective means for deciding when two labels are just two different words for the same thing, or whether they really name different things.

$IDEF_0$ labels are candidate $IDEF_1$ entity and attribute class names; if an entity class is used as an $IDEF_0$ arrow label, attributes of the entity class can aid in determining the precise arrow contents, and can be used to understand and compare arrows. Attribute listings for entities named in diagrams would be a useful addition to glossary formats. Given an attribute listing, "redundancy" could be assessed by the degree of overlap between sets of attributes.

### Definitions and Measures

| DEFINITION | PRECISION OF INFORMATION CONTAINED IN AND/OR CONVEYED BY A DIAGRAM OR MODEL; APPROPRIATENESS OF TERMINOLOGY AND SYMBOLOGY. ABSENCE OF INFORMATION PERIPHERAL TO MODEL'S ORIENTATION. |
|---|---|

| MEASURES AND PARAMETERS | • INFORMATION VALUE OF LABELS, TITLES, AND FUNCTION NAMES |
|---|---|
| | • DEGREE OF REDUNDANCIES IN LABEL NAMES |
| | • NATURALNESS OF TERMINOLOGY: TO FIT -- |
| | • SUBJECT MATTER |
| | • AUDIENCE |

## Consistency

The syntactic checks discussed earlier would serve to verify the graphic notations that help convey consistency in labelling and trace- ability of data arrows throughout the hierarchy of a model. The degree that the content of the model is consistent is more a matter of judgement on the part of reviewers and experts in the subject matter being modelled.

Consistency of naming, however, in the case of function box and arrow designations, might be facilitated if attribute listings were provided as part of glossaries with a diagram or model.

## Definitions and Measures

| DEFINITION | • INTERNAL: DEGREE OF UNIFORMITY OF NOTATION, SYMBOLOGY, AND TERMINOLOGY (WITHIN THE MODEL) |
|---|---|
| | • EXTERNAL: DEGREE THAT CONTENT IS TRACEABLE TO THE SYSTEM BEING MODELED. (OUTSIDE THE DIAGRAM OR MODEL) |
| MEASURES AND PARAMETERS | • CORRECTNESS OF ICOM LABELS: PLACEMENT AND TRACEABILITY THROUGH THE HIERARCHY |
| | • CORRECTNESS OF TUNNEL ( ) USE |

## Correctness

"Correctness" is perhaps the most subjective of the semantic characteristics presented in this section, chiefly because the final standard for assessment must come from the opinion of individuals who have the most complete knowledge of the subject matter being modelled. The kit cycle in $IDEF_0$ has been designed to assist the construction of models that can be judged "correct" on the basis of peer-review, as well as by evaluation by technical experts.

## Definitions and Measures

| DEFINITION | |
|---|---|
| | • EXTENT TO WHICH INFORMATION EXPRESSED IN A DIAGRAM OR MODEL IS AN <u>ACCURATE DESCRIPTION</u> OF THE SYSTEM BEING MODELED. |
| | • DEGREE TO WHICH IMPLIED CONSTRAINT RELATIONS AND DATA-TO-FUNCTION INTERFACES REPRESENT <u>VALID RELATIONS</u>. |

| MEASURES AND PARAMETERS | |
|---|---|
| | • REVIEW OF MODEL BY EXPERTS |
| | • CHECKS IN COMPANY LITERATURE OR TEXTBOOK SOURCES FOR TERMINOLOGY AND TECHNICAL USAGE. |

Complexity/Understandability

Understandability' is an important but difficult criteria to measure in assessing model quality, mainly because of its abstract and subjective connotations. Another source of the problems associated with measuring "understandability" is the strong interrelationship of syntactic and semantic factors. Since we are defining "understandability" as a function of how well the content of the model is portrayed via the syntax, the relevant measures would be derived from both syntax and semantics. No single feature of semantics would therefore appear to be a sufficient index of how well an evaluator could understand a complete model.

Measuring "complexity," however, is a more feasible task. We are suggesting that "understandability" and "complexity" be viewed as dual measures - i.e., something that is highly complex can be expected to be hard to understand; and similarly, something that has a low complexity value is probably much more accessible to a user and more easily understood.

Definitions and Measures

| | |
|---|---|
| **DEFINITION** | • EXTENT TO WHICH PURPOSE OF MODEL OR DIAGRAM IS CLEAR TO THE EVALUATOR. |
| | • DEGREE TO WHICH "WHAT THE MODEL/DIAGRAM SAYS" IS ACCURATELY DEPICTED VIA THE SYNTAX. |
| **MEASURES AND PARAMETERS** | • EASE OF FINDING THE "MAIN PATH;" |
| | • NUMBER OF ACTIVATIONS/BOX; |
| | • NUMBER OF POTENTIAL SCENARIOS; |
| | • FACILITY OF IDENTIFYING ACTIVATIONS AND SCENARIOS; |
| | • EASE AND SUCCESS OF "REFINING THE LAYOUT;" |
| | • COUPLING AND COHESION RATING SCHEME |

Another measure that would be useful to $IDEF_0$ authors and commenters is ease with which the layout on a diagram can be refined. "Refining the layout" is defined here as the procedures that are necessary to disambiguate the message of a diagram or model, as it is conveyed by the pattern of the function box and data-arrow interfaces. As a model becomes more and more complex, and thereby less and less understandable, the task of refining the layout becomes more complicated. Rules for refining the layout of a diagram to reduce its complexity are presented as follows:

- Boundary arrows with more than one ICOM code must be split so that each ICOM has its own arrow.
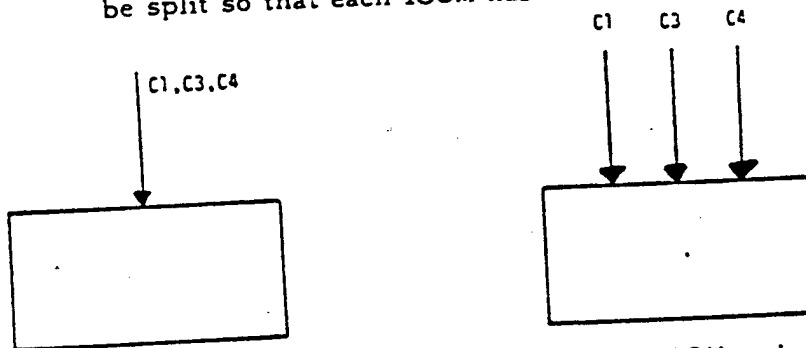


Figure 6-2. Arrows with more than one ICOM code

- Boundary arrows having the same ICOM codes must be connected.

BEFORE                                                    AFTER
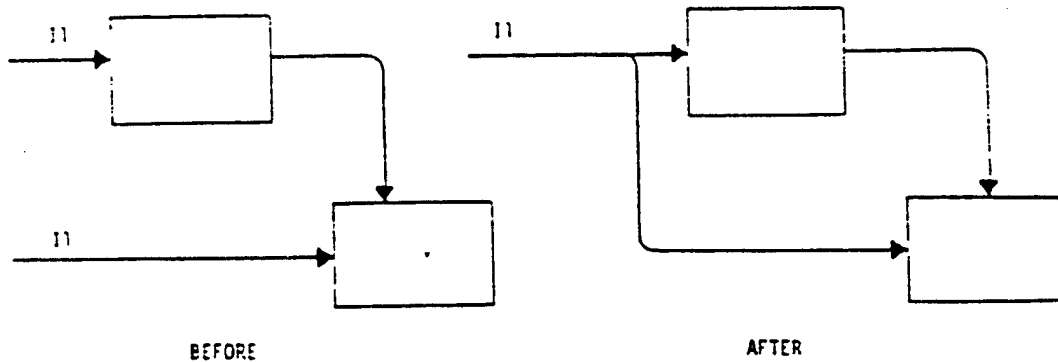
Figure 6-3.  Connecting Boundary Arrows

- Arrow names that change from parent diagram to child diagram, but still represent the same data, must be reconciled by the author.  The appropriate names must be determined by the author and used consistently throughout the model.
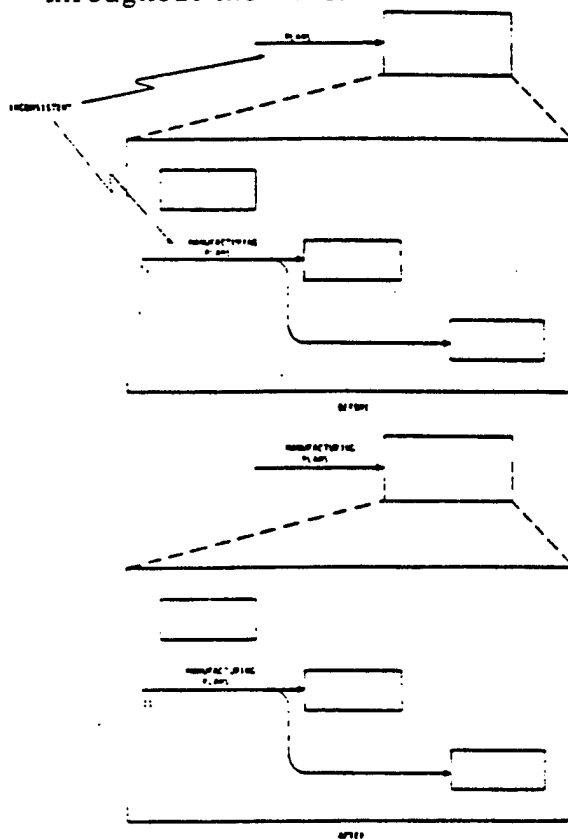
Figure 6-4.  Arrow Labels Remain the Same

- Arrow names that are the same but are intended to have different meanings must be assigned unique names by author.
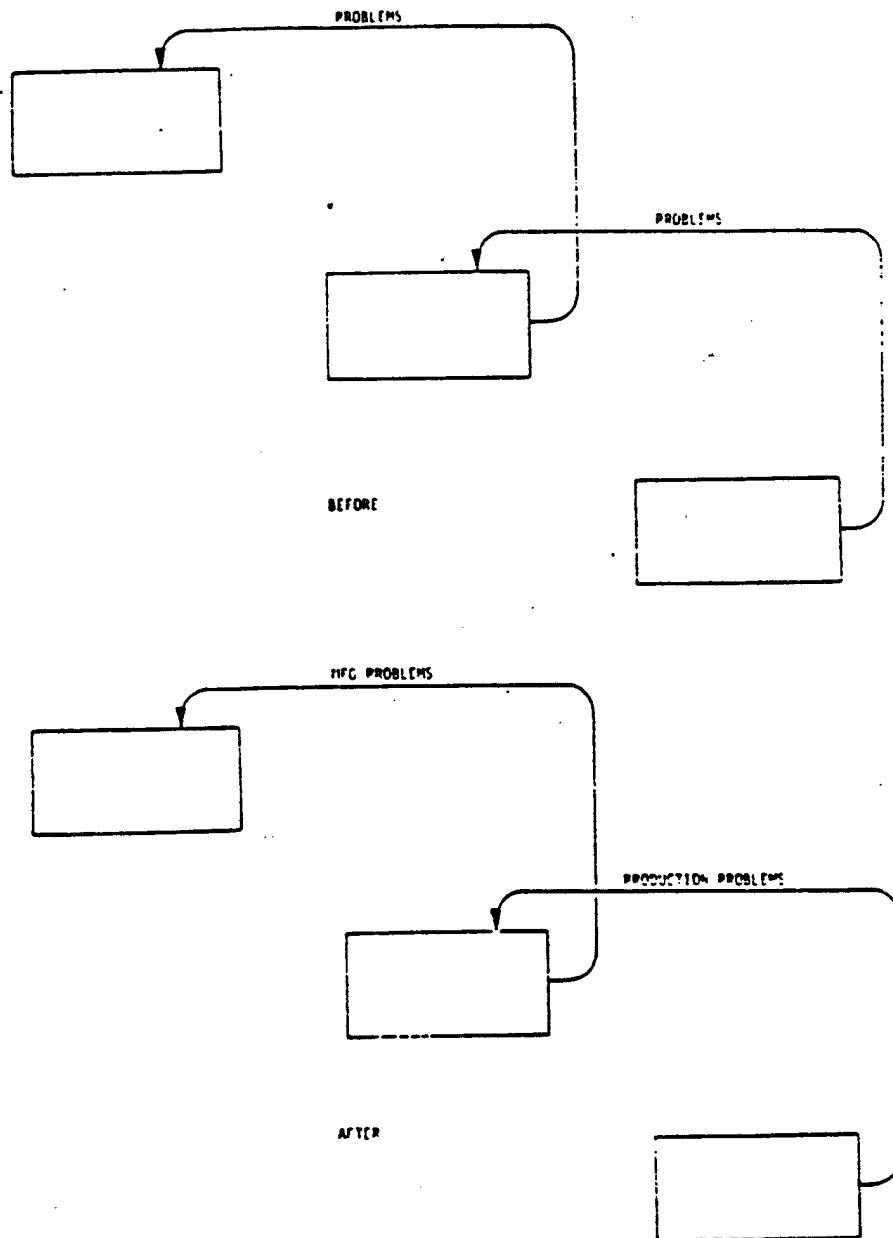


Figure 6-5.  Arrow Labels Defined

o  When dealing with arrows that split and join, any
unnamed line segments must be explicitly named by
author.



The name of this segment can be
deduced to be "products," so that
the explicit naming of the arrow
is not necessary, unless the
decomposition of box 1 shows
the arrow to have a name
different from "products."

The name of this segment
cannot be easily deduced without
understanding of the model (it
could be A, or D, or part of
both).  The name must be
specified by the author.
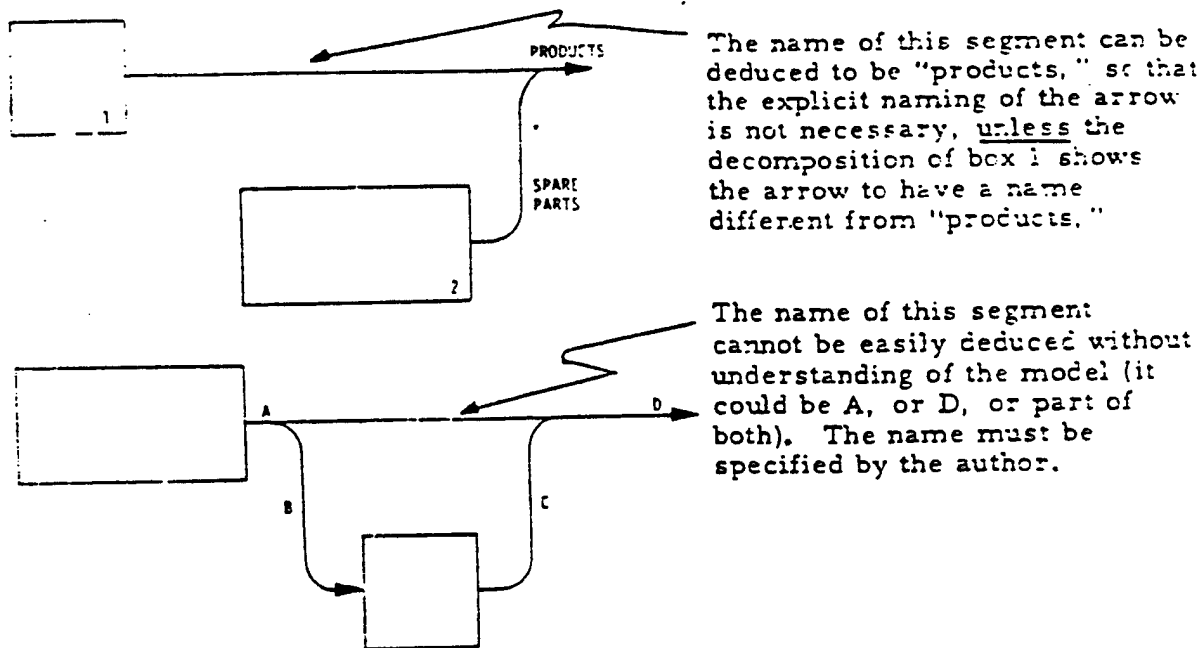
Figure 6-6.  Label Arrow Splits

### 6.6.3 Relationship Between Coupling and Cohesion

Coupling describes the degree of interconnectedness among components. For $IDEF_0$, one could describe the number and types of connections between and among function boxes on a single diagram as interconnections among components. Cohesion describes the degree of relatedness of subparts within a given part. For $IDEF_0$, "cohesion" would therefore be a meaningful property to use in evaluating the manner in which functions are grouped together in a single diagram.

The relevancy of these measures to assessing complexity is as follows:

"When components of a diagram or model are highly coupled to each other, the model is more complex. When components are composed of unrelated (incohesive) elements, the model is more complex."

Thus, $IDEF_0$ diagrams should become more understandable if their components are loosely coupled, but fall into functionally relatable and cohesive patterns.

### 6.6.4 Metrics Based on Coupling and Cohesion

The understandability of a model increases and complexity decreases as cohesion increases and coupling decreases. It follows that metrics based on types of coupling and cohesion will be useful for assessing the understandability/complexity of $IDEF_0$ models. Having a scale or code to identify different types of coupling and cohesion will also provide model authors with a tool that they can use in constructing models which are more understandable and less complex.

### 6.6.4.1 Relation to Other Systems Engineering Properties

There are several additional advantages that can be gained by establishing a metric to assess the types of coupling and cohesion in IDEF models. In particular, it will be easier to transition an "AS IS" system description to a "TO BE" status if coupling is reduced, thereby

restricting the number of module connections so that as few modules as possible depend directly on the same assumptions.*

The systems engineering principle of <u>hiding</u> describes the same situation: information known only to a given set of modules is said to be <u>hidden</u> (or isolated) from other modules. The advantage of information hiding, like reduced coupling, is to facilitate local changes in a model or a system design, without having to change as many modules or components as might otherwise be necessary. There is, then, a direct relationship between hiding and coupling, and an assessment of the degree of coupling within a particular model will provide information about the extent of information hiding as well.

Just as hiding appears to be isomorphic to coupling, <u>localization</u> is isomorphic to cohesion. Localization is a principle describing the value of bringing elements together in physical proximity -- the benefit of this principle is to minimize redundancies that might otherwise occur if the same information has to be stated over again in many scattered sites. For example, consider the potential value of identifying generic manufacturing subsystems from the ICAM "AS IS" Architecture (Function Model of Manufacturing). Localizing those functions relating to manufacturing control and materials management in a separate function model, for instance, will facilitate a transition to a "TO BE" environment because the relevant functions and data have been put into functional and physical proximity, and can therefore be viewed as a separate, but integral unit. When changes in the environment are anticipated, it would be considerably more difficult to have to trace all possible side effects on functions that are dispersed throughout the Architecture.

---

*Where "module connections" can be interpreted as the function/ data interfaces on an $IDEF_0$ diagram.

## 6.6.4.2 Measures and Types of Coupling

For purposes of this discussion, differences in types of coupling will be illustrated by examining connections between functions in $IDEF_0$.

### 6.6.4.2.1 Viewpoint of the Description

Traditionally in the systems engineering literature, the coupling between functions can be analyzed from two points of view: the first considers the nature of the data in the connection, and the second identifies the relations among many tokens or examples of data in a connection.

Figure 6-7 below displays this dual viewpoint analysis. The terms used under each category are those found in the literature, and their technical meaning will be treated in the following pages.

Nature of Connection                    Structure of the Connection

|  |  |  |  |
|---|---|---|---|
|  | Loosely Coupled |  |  |
| Normal | ↑ |  | Abstract |
| Control | Better | Worse | Record |
| Pathological | ↓ |  | Environmental |
|  | Tightly Coupled |  |  |
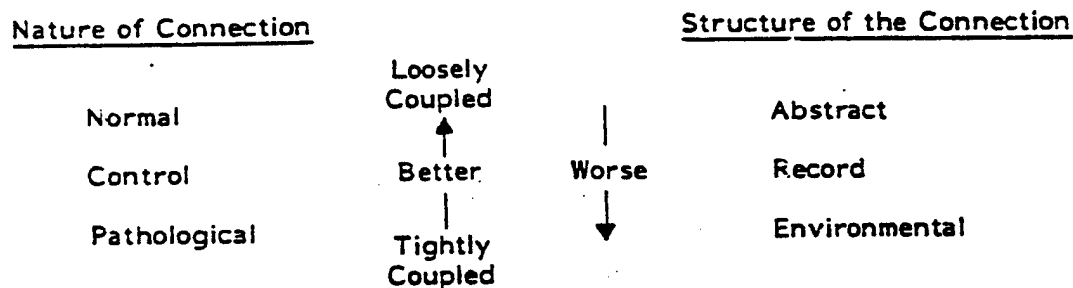
Figure 6-7. Analysis of Coupling Between Functions

### 6.6.4.2.2 Nature of Connections

#### Pathological: The Least Desirable

Two functions are said to be "pathologically coupled" if one function references the contents of the other. This situation occurs when the hiding principle has failed to apply. As an example, consider Figure 6-8.
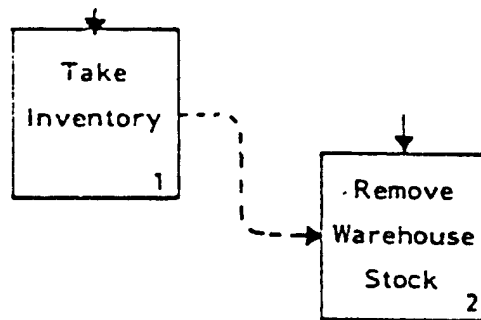
Figure 6-8. Pathological Coupling

A data arrow going between boxes 1 and 2 is "pathological" and the boxes are "content-coupled" by this definition. Clearly, the effect of box 2 removes the required materials that the activity in box 1 needs to operate with, and connections of this type are usually not desirable, unless other factors supply qualifications.

### Control

Two functions are "control coupled" if data from one influences the flow of control of the other. Flow-of-control is to be distinguished from a control-arrow on a box. "Control" in this context refers to the kind of data, not its position on a box. Control coupling occurs when a function generates a controlling effect on another function's behavior. This kind of coupling can be spotted by noting function-names that produce "flags" along with an intended functions, such as "Record failure/ print error message."

### Normal

Two functions are "normally" coupled if the data connections between them exist independently of the contents of either function, or if there is no influence on the flow-of-control from one function to the other. This type of coupling is the most desirable kind of connection, since communication between functions is via explicitly-shown data, and not via implicit references to the contents of one or the other functions.

## 6.6.4.2.3 Structure of the Connection

### Environmental

Two functions are environmentally coupled if they both reference a generic data source, without clearly specifying the relation of a particular piece of data to any box. This situation can be represented by the diagrams in Figure 6-9.
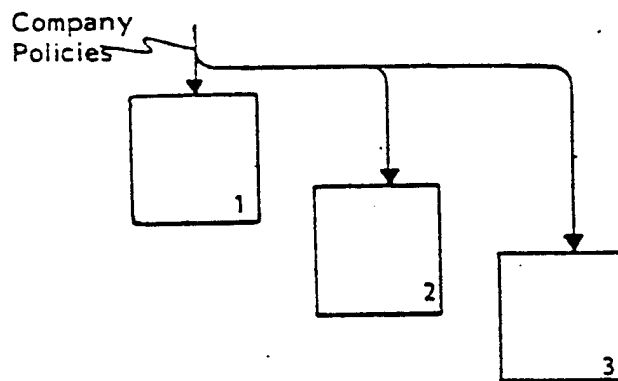
Figure 6-9. Environmental Coupling

Problems resulting from environmental coupling include the following:

1)  The hiding principle is violated. Access is not restricted to only the necessary data.

2)  Localization is violated. Data is not associated only with its users.

3)  Error propagation is increased, potentially. Changes to the environment may affect all functions unpredictably.

## Record

Two or more functions are record-coupled if they reference a non-global set of data, and do not necessarily depend on having all possible interfaces displayed, as in environmental coupling. An $IDEF_0$ example appears in Figure 6-10.
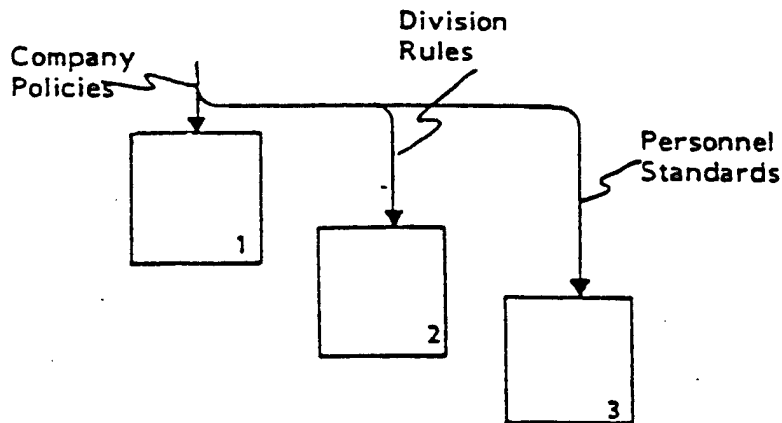


Figure 6-10. Record Coupling

The type of interface pattern shown here is preferred because the relation of a particular data element to each box has been specified.

## Abstract

Two elements have abstract coupling if the relation between a particular element and its potential "constituents" has been clearly specified. In $IDEF_0$, a data arrow on a parent diagram that gets broken down into detail components on child diagrams represents an abstract coupling relation. Hiding is in effect, insofar as the parent-level arrow is "using the power of abstraction" to provide access to the level of abstraction required for more complete understanding of the data contents. The notion of "arrow pipelines" in $IDEF_0$ corresponds to this use of abstract coupling, and should be maximized when constructing diagrams.

### 6.6.5 Measures and Types of Cohesion

Whereas "coupling" refers to the connections between elements (functions or data), "cohesion" will be used to reference the binding or internal-relatedness between factors on a diagram.

At least seven types of binding have been distinguished in the literature.

| Relative Value | | Worse |
|---|---|---|
| 0 | Coincidental | |
| 1 | Logical | |
| 2 | Temporal | |
| 3 | Procedural | |
| 4 | Communicational | |
| 5 | Sequential | |
| 6 | Functional | Better |

Each type will be identified briefly, and illustrated by way of a sketch of a typical example in $IDEF_0$.

### (û) Coincidental:  Least Desirable

Coincidental cohesion occurs when there is little or no constructive relationship among the elements of a module.  This situation obtains when the data names on $IDEF_0$ arrows in a single diagram bear little relation to each other.  The extreme version of this case is shown in Figure 6-11.
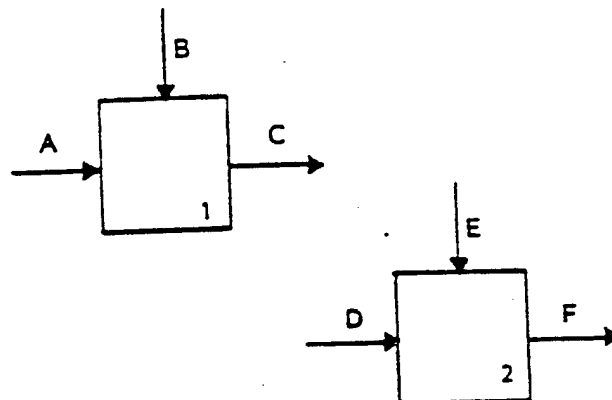
Figure 6-11.  Coincidental Binding

6-45

(1) Logical

Logical binding occurs when the data and functions seem to have been gathered because they fit into a common class or set of elements -- but no necessary functional relation can usually be identified.

Figure 6-12 illustrates the typical cause of logical binding.
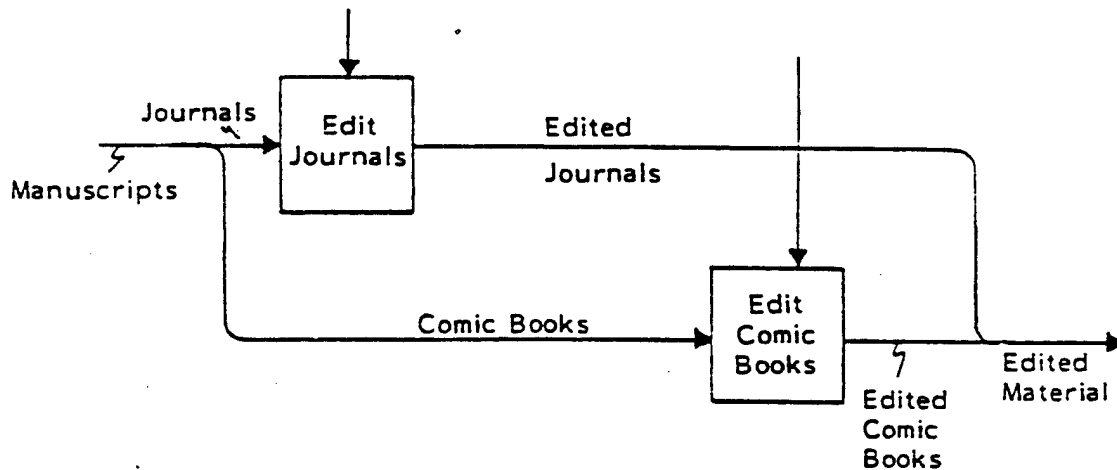


Figure 6-12. Logical Binding

(2) Temporal

Temporally bound elements appear to be clustered because they represent functions associated in time -- data are used simultaneously, or functions are performed in parallel, rather than in series. A frequent example can be found in diagrams showing functions all concerned with "initialize" operations, as pictured in Figure 6-13.
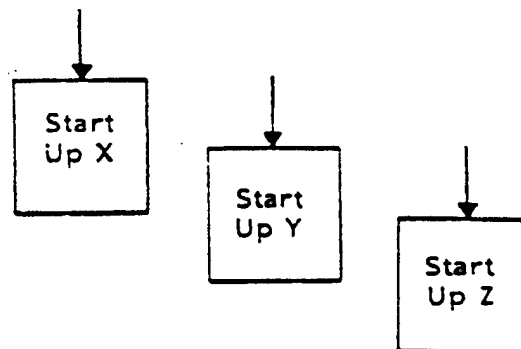


Figure 6-13. Temporal Binding

### (3) Procedural

Procedurally-bound elements appear grouped together because they are executed or performed during the same part of a cycle or process. The common unit of process may be an iteration or decision-branch, or a linear ordering of steps. An example of a procedurally-bound diagram is given in Figure 6-14.
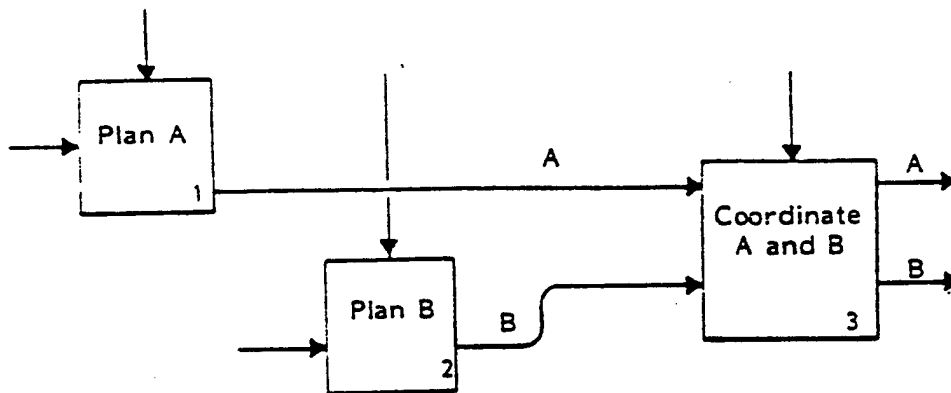


Figure 6-14. Procedural Binding

In this diagram, the boxes are clustered because both outputs, A & B, are necessary before some function downstream can proceed. Note that boxes 1 and 2 have no other influence on each other, and they are thereby bound by their relation to the process in box 3.

### (4) Communicational

Diagrams exhibit communicational binding when boxes have been grouped because they use the same input data and/or produce the same output data. This is the first type of binding discussed thusfar which represents the _preferred_ level of binding for $IDEF_0$. One important reason is that none of the levels of cohesion discussed above is very closely tied to the structure of a particular problem. _Communicational cohesion_ is the lowest level at which we encounter a relationship among processing elements which is intrinsically problem-dependent.

Figure 6-15 illustrates the typical case.
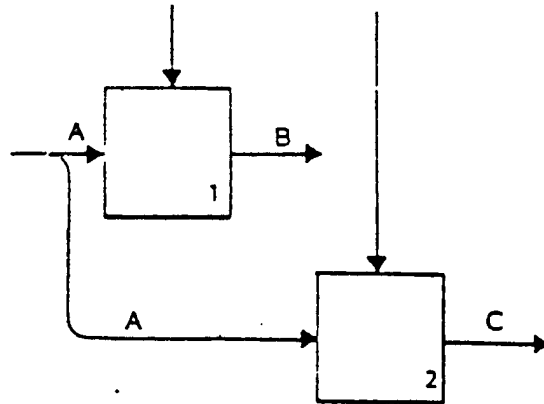
Figure 6-15.  Communicational Binding

## (5) Sequential

Diagrams having sequential binding have output from one function
serving as input data for the next function.  The relationship among
elements on a diagram is more interrelated than at previously-mentioned
binding levels, insofar as cause-and-effect transforms of data are being
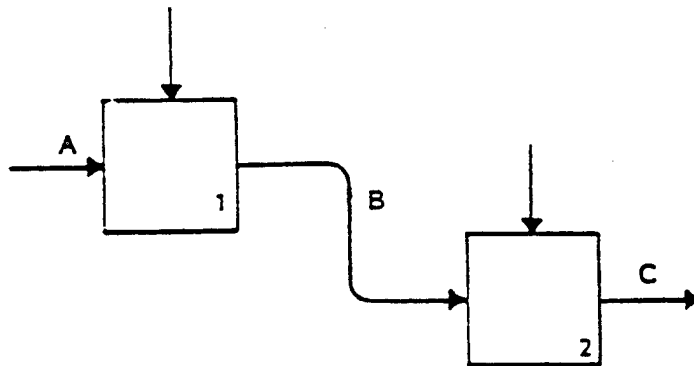approximated.  Hence, the high rating.  Figure 6-16 illustrates sequential
binding.

Figure 6-16.  Sequential Binding

(6) Functional

A diagram exhibits complete functional binding when all elements of a function contribute to the performance of a single function or result. A diagram that is purely functional contains no extraneous elements related by sequential or weaker-valued types of binding. One way to identify functionally-bound diagrams is to look for two boxes linked via control arrows, as illustrated in Figure 6-17.
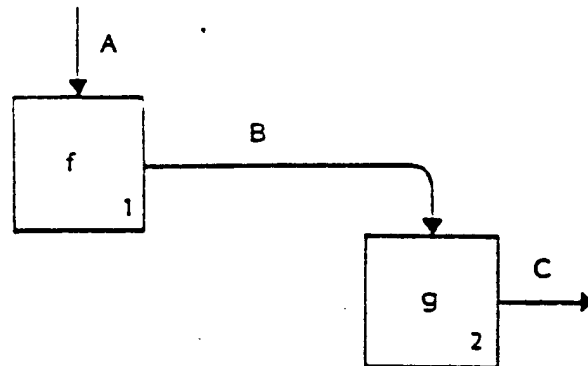


Figure 6-17. Functional Binding

In mathematical terms, the requirement for the simplest type of functional binding, shown in Figure 6-17, is as follows:

$$C = g(b) = g(f(A))$$

Other useful checks for functional binding include clues-by-elimination. The guidelines that follow are meant as aids to help authors and reviewers distinguish functional from non-functional binding.

- If the only reasonable way of describing a diagram is by using a compound sentence, or a sentence containing a comma, or a sentence containing more than one verb, then the diagram is probably less than functional. It is probably <u>sequential</u>, <u>communicational</u>, or <u>logical</u> in terms of cohesiveness.

- If the descriptive sentence contains such time-oriented words as "first," "next," "after," "then," "start," "step," "when," "until," or "for all," then the diagram probably has temporal or procedural cohesion, sometimes, but less often, such words are indicative of sequential cohesion.

- If the predicate of the descriptive sentence does not contain a single specific object following the verb, the diagram is probably logically cohesive.

## Summary

Figure 6-18 contains a summary of the types of binding treated in the previous sections. The important point to note is that levels 4-6 constitute the types of binding that $IDEF_0$ developers consider critical for having "quality" diagrams. Authors should strive to maximize the presence of these binding-types.

## 6.6.6  Assessing Coupling and Cohesion in IDEF

The rating scales presented here were originally developed to assist systems engineers in selecting among design alternatives, and to clarify the relative desirability of alternate decompositions. The use of similar scales is being proposed for $IDEF_0$ as an aid to authors and commenters for ensuring the quality of models.

A few additional observations are in order, in the light of these objectives:

- With relevance to $IDEF_0$, "coupling" and "binding" are really versions of the same concept, but differ because they apply to distinct components of models. "Coupling" can be used to compare the complexity of connections between diagrams, or between boxes on a parent diagram.
  "Binding" can be used to assess the strength of the links among different boxes within a diagram, or to measure the cohesiveness of the parent box of a diagram.

| Rating | Level of Binding | For Functions — Random | For Data — Random |
|---|---|---|---|
| 0 | Coincidental | Random | Random |
| 1 | Logical | Functions of the same set or type (e.g., "edit all input") | Data of the same set or type. |
| 2 | Temporal | Functions of the same time period. (e.g., "initialize operations") | Date used during a time period. |
| 3 | Procedural | Functions occuring in same phase or iteration. (e.g., first pass of a compiler) | Data used during same phase or iteration. |
| 4 | Communicational | Functions that use the same data. | Data acted upon by the same activity. |
| 5 | Sequential | Functions that perform sequential transforms of the same data. | Data transformed by sequential functions. |
| 6 | Functional | Functions that combine to perform one function. | Data associated with a single function. |

Most Desirable

Figure 6-18. Cohesion: Levels of Binding

- Immediate clues for identifying undesirable connections can be found by noticing tight coupling on a parent diagram. This usually indicates low-valued binding (incohesion) on child diagrams. An author can improve the quality of the model -- and make the set of parent:child diagrams more understandable by re-decomposing the parent to eliminate the complex coupling.

- Rating values for coupling and binding are additive. Many connections meet the definitions for more than one category of the rating scale, and can accumulate multiple values. For example, if a diagram is both "sequentially" and "communicationally" bound, it is _better_ in binding quality than being strictly communicational. But if a connection is both "control" and "record" - coupled, it is _worse_ (and more tightly coupled) than being simply control or record - coupled alone.

SECTION 7

## DATA COLLECTION FOR
## IDEF MODELING

### 7.1 Introduction

When analyzing or designing any system, it may be necessary to obtain or verify facts about the system or subject matter at hand. There are many sources of factual information. One might:

- read existing documents, using each table of contents and index to locate needed information.

- observe the system in operation, if it already exists.

- survey a large group of people, through question-naires or other such means.

- talk to one or more "experts" who possess the desired knowledge.

- use whatever is already known by the author.

- guess or invent a hypothetical description, and ask readers to help bring it closer to reality.

Of all these methods, the most important is face-to-face interaction with an expert. Seldom will all existing information be written. Preconceived notions that are reflected in questionnaires are often faulty.

Obtaining information from an expert has been formalized in an interview process. This step by step process allows an interview to be conducted without unduly influencing the expert with information already obtained by the interviewers.

A key part of interviewing is to record the information obtained. This can be done either as informal notes, as activity and data lists, as a formal matrix of functions, or as diagram sketches.

## 7.2    The Interview Process

The purpose of an interview is to gather information from an individual who possesses an expertise considered important to the analytical effort.  There are four types of interviews that might be conducted during the course of performing the analysis phase of an IDEF project.

### Four Types of Interviews

a)    Fact finding for understanding current operations. This type of interview is used to establish the content of a Current Operations Model, or to help understand the existing environment.

b)    Problem Identification to assist in the establishment of future requirements.  This type of interview is used to validate the Current Operations Model and to provide the foundation for a Future Operations Model.

c)    Solution Discussion regarding future system capabilities.  This type of interview is used to establish the content of a Future Operations Model.

d)    IDEF Author/Commenter Talk Session.  This type of interview is used to resolve problems which have surfaced during the construction of an IDEF model.

The reason for identifying types of interviews is that during the course of performing an actual interview, ingredients of each type appear. The respondent might tell the interviewer facts about a given system in terms of problems.  Also, the respondent might identify problems in terms of solutions to the problems.  The interviewer, by constantly classifying the respondants remarks, can obtain the maximum useful information from the interview.

7.3    The Interview Kit

It is recommended that a "standard" Interview Kit be used for recording the interview. It may be stored in an Interview File and it may be distributed to appropriate individuals. This distribution might include other members on the Analysis team or even the interview respondent for corrections, additions, and deletions. The interview kit would contain:

1.    Cover Page (Kit cover)

2.    Interview and Record Follow-up

    a.    Interviewer Name (IDEF Author Name)

    b.    Interview Date (IDEF Diagram Date)

    c)    Interview Duration (Start time, End time)

    d)    Respondent Name

    e).    Respondent Title and Organizational Responsibility

    f)    Respondent Telephone Number and Extension

    g)    Additional Sources of Information Identified

        1)    Documents - Title and Location

        2)    Other Interviewees - Name, Title,
            Organizational Responsibility,
            Address, Telephone number

    h)    Essential Elements of Information - A Summary
       of the key points covered in the interview.

    i)    Follow-up questions and/or areas of concern
       either not covered during the interview, or
       postponed

    j)    New Terms for Project Glossary

3.    Activity and Data List

4.    Interview Agenda (Developed in preparation of
     Interview - This is covered in following section)

5.    Interview Notes and Rough Diagrams

## 7.4    Interview Guidelines

There are five stages to the successful interview; each must be performed in order to assure that the most information is obtained and recorded in the least amount of time.

- Preparation

- Initialization

- Interview

- Termination

- Finalization

In each stage of an interview there are certain basic activities which must be performed. Additionally, associated with each stage, there exist psychological aids which will help the interviewer establish an atmosphere of professionalism and trust with the respondent.

### 7.4.1  Interview Preparation

By thinking through certain key interview needs before the interview, a more organized and efficient dialogue can be achieved. Preparation for an interview should contain, but is not limited to, the following activities:

1.    Select Interviewee

    a)    From areas of responsibility

    b)    From recommendations of others

    c)    From various levels of the organizational hierarchy - upper levels useful for "big picture," lower levels for detail information, and middle levels for bridging the gap

2.    Make Appointment

    a)    Short duration - $\frac{1}{2}$ to 1 hour

    b)    Not immediately before lunch, nor late afternoon

    c)    Identify purpose of interview

    d)    Explain interviewer role

3. Establish Tentative Agenda

    a) Topical areas - used as a foundation for interview (this helps prepare "broad general questions")

    b) Specific questions

4. Review applicable background information

5. Review appropriate terminology

6. Insure coordination with other interviews

    Check interview file to ascertain that the respondent has or has not been previously interviewed. If the interview is a follow-up interview, then examine the results of previous interviews.

7. Fill out Interview Record and Follow-up with pertinent information

8. Make out Interview Agenda

## 7.4.2 Interview Initialization

This stage of the interview is directed at establishing a rapport between the interviewer and respondent. The courtesy permitted by the respondent at the start of an interview is usually short. This time is important in motivating the respondent to help the interviewer. This stage of the interview should contain the following topics:

1. Provide respondent with a tangible means of introduction, e.g., a business card (this removes doubt on the part of the respondent as to how to pronounce or spell the interviewer's name and can therefore remove a frequent cause of respondent embarrassment)

2. Establish purpose of interview

    a) Expand on information provided in initial contact

    b) Establish point of view for the interview. Use interview type 1, 2, 3 or 4 as a basis.

    c) Establish purpose of the interview - even if the interview is a follow-up interview.

3.   Establish the acceptability of note taking. The respondent may require assurance of confidentiality.

4.   Establish the Expert/Author relationship - alleviate the fear that the interview will be used to tell the respondent how to do his job, or that the respondent's job is in jeopardy.

5.   Start with broad general questions which will get the respondent talking - these should be based upon the topical areas identified in the agenda.

6.   Assess the respondent's ability to provide pertinent information - if the information is too general or too detailed for the stage of the IDEF model being prepared, evaluate respondent's ability to contribute. Terminate the interview if necessary - it may be a waste of both the interviewer's and respondent's time.

7.   Begin to formulate specific questions which complement the agenda.

8.   Write, don't talk.


## 7.4.3  Conducting the Interview

While it is not useful to define questions to ask during an interview, it is possible to identify guidelines that should be considered during the interview. The first set of guidelines deals with the qualification of the information being obtained. The second set of guidelines relate to the stimulation of information flow.

Information Qualification: The human mind can comprehend at double the rate at which people speak. The danger in interviewing is that this rate difference is typically used by the listener to think about what should be said in response instead of about what is being said.

To assist the interviewer in thinking about what is being said, there is a series of questions which may be used to help the interviewer keep his mind on the information being provided:

●   What supporting facts are being provided for the main points being discussed?

●   How recent is the information?

- How complete is the information?

- Do I really understand what is being said?

- Is the level of detail being presented appropriate for my purpose?

- Are there areas being omitted?

- Has this information been discussed with someone else?

- How important is this information?

- Are side-topics being discussed?

- Has the interview viewpoint changed?

Information Flow Stimulation: The following set of guidelines can be used to stimulate the respondent into providing maximum reliable information:

- Keep extraneous comments and conversations to a minimum. The interview is used to obtain information, not make friends, or sell ideas.

- Be aware of the respondent's failure to identify problem areas in the environment. This may indicate that the respondent is not at ease with the interviewer.

- Provide the respondent time to think. Do not suggest answers or ask another question. A pause in the interview is useful to allow the respondent to recall vital pieces of information.

- Avoid outside distractions which tend to "uncouple" the train of thought. If at all possible, conduct interviews outside of the respondent's normal habitat.

- Be aware of internal distractions, signs that the respondent is not comfortable or at ease with the interview.

- Try to determine if the information being obtained is fact or opinion.

- Encourage elaboration by asking for a rephrasing or a summary of the information presented.

- Ascertain the respondent's background and association with the subject matter being discussed. Valuable insight into the respondent's remarks can be obtained knowing his relationship to the organization and existing systems.

- Do not enter into or encourage sarcasm and humor.

- Do not mention or discuss any interview with another person.

- Record all questions asked by the respondent. The interviewer should answer all questions except those dealing with user-organization management, plans, or personalities.

- Show interest in what the respondent is saying.

- Concentrate on the unfamiliar and difficult aspects of the subject being discussed. Avoid the obvious.

- Be alert for the inconsistent or incorrect use of words. Ask for definitions for any unfamiliar or questionable term. Record the definition for the project glossary.

- Do not contradict the respondent even if facts do not support what is being said. Use the Kit Cycle to resolve such conflicts.

- Be humble. The respondent is the expert, not the interviewer.

- Postpone subjects which cannot be fully covered within the agreed upon time frame. Do not extend the interview time, but rather make another appointment.

- Appreciate different opinions on the same subject. Use IDEF to show these opinions and to resolve conflicts.

- Stimulate the respondent with pertinent open ended questions.

## 7.4.4 Termination

The interview should be terminated for any of the four following reasons:

a) The information being obtained in the interview is not appropriate.

b) The time limit has been reached.

c) The interviewer has been saturated with information.

d) There is a clash of personalities between the interviewer and the respondent.

Depending upon the cause of termination, the following topics should be considered during the termination of the interview:

- The interview should not be closed abruptly, but rather should end with a few minutes of informal discussion.

- The main points of the interview should be summarized.

- Areas of concern which have been postponed or not covered should be identified.

- A follow-up interview, if necessary, should be arranged.

- The respondent should be asked to recommend other persons who should be interviewed.

- If the interview notes are to be reviewed by the respondent prior to distribution, this fact should be mentioned during the termination.

- The respondent should be thanked for his time and effort.

## 7.4.5 Finalization

This stage of the interview is directed at assuring that the information obtained during the interview is properly recorded and disseminated to the project team. The vehicle used to accomplish the finalization of an interview is the Interview Kit. If note taking was not permitted by the respondent, the interviewer should, upon termination of the interview, immediately write down the salient points discussed. Finalization of the interview includes the following:

a) Identify additional sources of information.

b) Summarize the Essential Elements of Information.

c) Identify new terms for the project glossary.

d) List the follow-up questions and areas of concern either postponed or not covered during the interview.

e)  Complete Activity and Data List.

f)  Expand on any notes with any information recalled
during the review.

g)  Prepare rough IDEF diagrams that reflect the information
obtained.

h)  Identify any assumptions being made
or any items which are not clear.

i)  Publish and distribute the Interview Kit.

j)  Address the names, areas of expertise, phone
numbers, and addresses of persons mentioned
in the interview.

# APPENDIX A

# IDEF$_0$ GLOSSARY

CONTROL

INPUT → FUNCTION → OUTPUT

MECHANISM
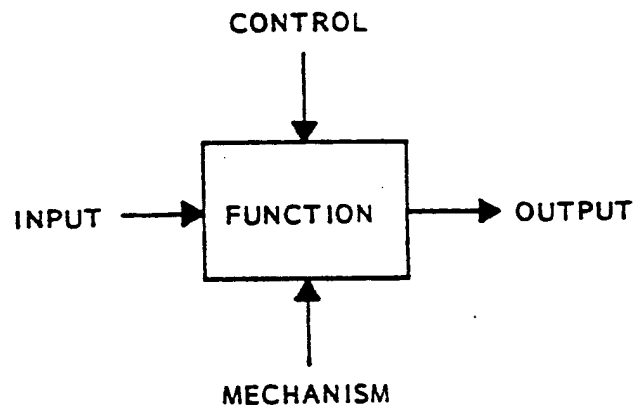
Arrow

A line representing data, its source (no point) and its use (point on the end of the line).

Author

The person who prepares any IDEF model.

Box

A rectangle, containing a name and number, used to represent an activity.

Branch

A fork or a join.

C-Number

A chronological number used near the lower right hand corner of an IDEF diagram form to: uniquely identify the diagram; trace the history and filing of an author's diagrams. C-numbers may be used as Detail Reference Expressions.

Call

A pointer (outward pointer on the bottom of a box) used to show that the box is detailed by the decomposition of another box.

Commenter

A person who has enough training in an IDEF technique to offer structured comments using the note numbering system and (often) referring to flaws in the application of the technique itself.

Context

The immediate environment in which a model is to operate; the limits of the model. In $IDEF_0$ the arrows around any box, but particularly the box on an A-O diagram. Also, the small box on the IDEF form in which the parent diagram and box are identified.

Control

The class of arrows associated with the top of an $IDEF_0$ box. Provides guidance to the transformation.

Data

Anything namable by a noun phrase such as things, conditions or information. Usually refers to a class (such as person) but may mean a single instance ("John Jones").

Detail Reference
Expression

The C-number or node number written beneath an $IDEF_0$ box to show that it is detailed and where.

Draft

An approval level for an IDEF diagram form above "working" and below "recommended."

A-2

| | |
|---|---|
| Expert | A person familiar with a part of the real world system being modeled. May serve as a source of information or as a reviewer of part of the model. |
| FEO | A diagram "For Exposition Only" in which violation of normal syntactic rules is allowed. |
| Fork | The point at which an $IDEF_0$ arrow (going from source to use) divides into two or more arrows. |
| Function | An activity described by a verb phrase that identifies what must be accomplished. |
| Glossary | A required section of an IDEF model which defines the way in which words or phrases are used. |
| ICOM | A single use of the ICOM code system. The acronym of Input, Control, Output, Mechanism. The arrows so labeled. |
| IDEF Role | A position in an IDEF project. See author, expert, commenter, reader, librarian. |
| Input | The arrow class associated with the left hand side of an $IDEF_0$ box. Usually becomes part of the output. |
| Join | The point at which an $IDEF_0$ arrow (going from source to use) joins with one or more other arrows to form a single arrow. |
| Kit | The standardized packages of diagrams which contain portions of, or complete to date, models to be reviewed. See kit cycle. |
| Kit Cycle | A formal procedure for obtaining peer or expert review during model development. |
| Label | The name associated with an $IDEF_0$ arrow. |
| Librarian | The person responsible for: <ul><li>routing and tracking of kits</li><li>project files</li></ul> |
| Mechanism | The arrow class associated with the bottoms of $IDEF_0$ boxes. |

Model                 A representation of a system which can be used to answer questions about the system.

Modeler               An alternate term for author.

Node                  A point at which subsidiary parts originate or center. The number associated with an $IDEF_0$ box or diagram. (Each activity may be shown once as a box and once as a diagram.)

Node List             A listing, often indented, showing all nodes in an $IDEF_0$ model in "outline" order.

Node Diagram          A graphic representation of the relationship between the nodes of an $IDEF_0$ model.

Note                  A comment on an IDEF diagram to record a fact outside those normally treated by the method or a comment by a reader or commenter about a diagram.

Output                The class of arrows associated with the right hand side of the $IDEF_0$ boxes. The result of an $IDEF_0$ transformation.

Parent                The diagram on which the box appears which is detailed by the "offspring" diagram.

Project               The organized task for which an IDEF model is prepared.

Project Manager       The member of the project who has final responsibility for the finished product.

Publication           The highest approval level for an IDEF diagram.

Purpose               A brief statement of the use to be made of a model so that the reason for its existence is clear.

Reader                A person with no, or limited, training in an IDEF technique who sees part or all of the model. A reader will often comment, but his comments are not expected to be structured. Individuals or groups participating in a walkthrough of a diagram are normally grouped as "readers."

Recommended           The next-to-highest approval level for an IDEF diagram.

A-1

Technical Committee    The group authorized to guide the development of a model and, eventually, to approve its contents.

Text    An overall verbal comment on an $IDEF_0$ diagram appearing on a separate diagram form.

Tunnelled Arrow    An $IDEF_0$ arrow one end of which is not associated with an arrow on the parent or offspring diagram.

Viewpoint    An attempt to define the subset of possible facts within a context which will be portrayed. Often expressed in terms of the persons whose perceptions are portrayed.

Working    The lowest approval level for an IDEF diagram. All IDEF diagrams are initially classified "working."

# APPENDIX B

## IDEF$_0$ INDEX OF TERMS

CONTROL

INPUT → FUNCTION → OUTPUT

MECHANISM

# IDEF$_0$ INDEX OF TERMS

# INDEX (Continued)

## INDEX (Continued)

## INDEX (Continued)

## INDEX (Continued)

# INDEX (Continued)